ROBOTICS ■ MICROCONTROLLERS ■ COMPUTER CONTROL ■ SPACE

NUTS AND VOLTS

EVERYTHING FOR ELECTRONICS

www.nutsvolts.com
SEPTEMBER 2006

HIGH VOLTAGE PIC

Generate 200 Volts DC From a Five-Volt DC Power Supply

Nine Volt Function Generator

Build this Period Counter/Totalizer

More on Digital Logic ICs

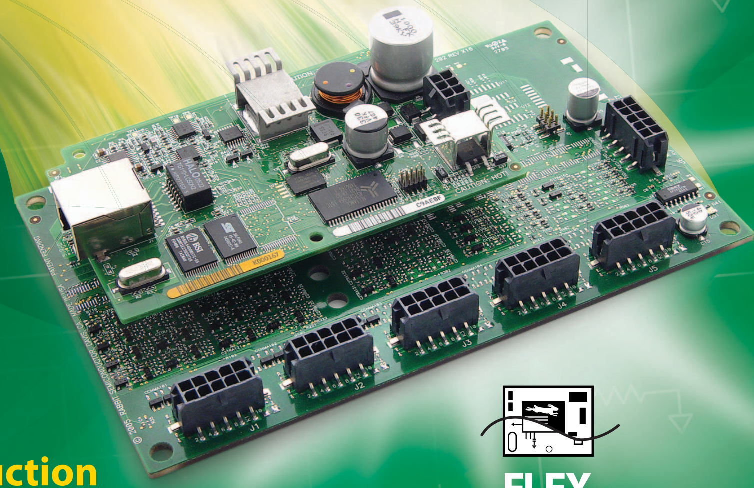
SX/B Turns *Sweet 16*

More Than You Ever Wanted To Know About TCP/IP


THESE ARE THE REAL DEAL
SerialLCD™
2X16 and 4X20
Serial Displays
RS232 and TTL Inputs
9600/2400 Baud
254 Backlight levels
RoHS Compliant
Input Serial Data
Normal/Inverted
Outputs for
Buzzer or Relay
Easy to use!
Priced from $34.95
Complete info
at BasicX.com
COLOR MAKES THE ORDINARY... EXTRAORDINARY!
NetMedia
NetMedia Inc., 10940 N Stallard Pl, Tucson Arizona 85737 tel.520.544.4567 fax.520.544.0800 www.BasicX.com

RabbitFLEX

A New Way To Customize



- **Click-to-ship in 5 days!**
- **Pay only for what you need**
- **Revision friendly**
- **Perfect for prototype and production**

FLEX Ready

RabbitFLEX™ is an unique build system that gives you the power to develop custom boards without the hassle and the cost. The RabbitFLEX simple-to-use web interface allows you to choose from numerous options such as digital I/O, analog I/O, serial ports, and Ethernet connections on your custom board. Just configure and buy online and our patent pending manufacturing process will deliver your solution in a matter of days. With RabbitFLEX you will reduce design risk, manufacturing cost, and development time.

Start developing now by ordering the RabbitFLEX Tool Kit and your own custom RabbitFLEX board. Take your solution to the next level.

Configure and Buy Online
www.myrabbitflex.com

Quick-Turn Boards Range From	RabbitFLEX Tool Kit
$149-$279	$199

Test Drive RabbitFLEX

Build your custom RabbitFLEX board online.
Add a tool kit to your order for a complete development system including Dynamic C®.

www.myrabbitflex.com

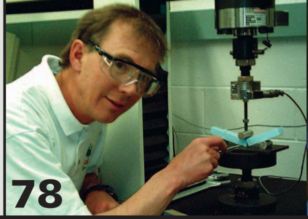
$49 Value
Free Book

For a limited time with kit purchase.

RABBIT® Semiconductor

2900 Spafford Street, Davis, CA 95616 Tel **530.757.8400**
Solutions That Work

06087b



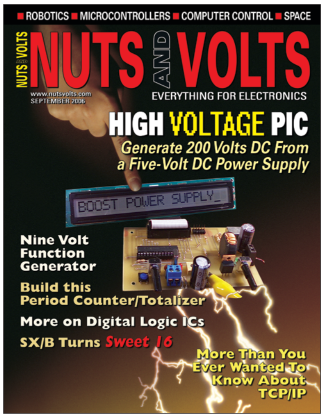
NUTS AND VOLTS

SEPTEMBER 2006

PROJECTS and FEATURES

ON THE COVER ...

Can you generate 200 volts DC from a five-volt DC power supply? You bet you can!



COLUMNS

DEPARTMENTS

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree#40702530) is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to **Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@nutsvolts.com

READER FEEDBACK

ELF-ISH DELIGHT

I couldn't have been more pleased when my current issue of *Nuts & Volts* arrived with the article about reviving the Elf. It was that very article in *Radio Electronics* in the mid-70s that really sparked my career in computers. Fascination began after wire wrapping and learning how one million things were happening every second within the CDP1802. Inputting programs, then losing them at power-off was soon overcome when they published the serial program whereupon they could be saved on an ordinary cassette tape. This all led to laying out a circuit board for my Elf; designing and laying out in CAD an 8K memory card (WOW!) and on and on ad infinitum. Displaying this at our company picnic moved me from maintenance to a design engineering job where I pursued a 24 year CAD designing career moving from company to company. I also praised the Lord that my wife "kept me on" after years of invading her kitchen sink with etching solutions. Keep up your wonderful work *Nuts & Volts*. P.S. I STILL HAVE the working model — it's part of my heart.

Fredrick L. Howe

HOT ALTERNATIVE

I am a long time electronics hobbyist, but a newcomer to your magazine. After receiving a subscription as a gift a few months ago, I have been enjoying your magazine each and every month. After reading your recent article on heatsink selection, as well as repeated questions (and interesting solutions) from readers in your Forum column, I am inspired to share with your readers an alternative to the venerable 7805 and heatsink for +5 volt supplies we often need for our many digital projects. A few years ago while designing a PIC-based LED clock, I was faced with the age old problem of power supply design. I wanted to use an AC-output wall transformer so I could get a zero-crossing line frequency reference to the PIC. This meant that the majority of my power supply would reside on-board, including the voltage regulator for the digital circuitry. Using a 7805 was an option, but with an input voltage of somewhere between 9 and 15 volts, the heatsink required a great deal of board real estate, not to mention the generated heat. Furthermore, choosing a wall transformer with the proper output voltage and current ratings to work under the varying load conditions my circuit would present,

Continued on page 98

by J. Shuman

NUTS AND VOLTS
EVERYTHING FOR ELECTRONICS

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct.
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Product Order Line **1-800-783-4624**
www.nutsvolts.com

Subscriptions
Inside US **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

FOUNDER/ASSOCIATE PUBLISHER
Jack Lemieux

PUBLISHER
Larry Lemieux
publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@nutsvolts.com

CONTRIBUTING EDITORS
Chuck Hellebuyck, TJ Byers, Jeff Eckert, Jon Williams, Peter Best, Phil Davis/Ken Tait, Dave Prochnow, Paul Verhage, Ray Marston, Robert Lang, Frank Reiser, Robert Reed, Michael Simpson, Robert Armstrong

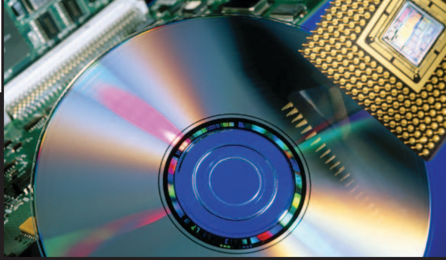
CIRCULATION DIRECTOR
Tracy Kerley
subscribe@nutsvolts.com

SHOW COORDINATOR
Audrey Lemieux

WEB CONTENT/NV STORE
Michael Kaudze
sales@nutsvolts.com

PRODUCTION/GRAPHICS
Shannon Lemieux

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher



TECH KNOWLEDGEY 2006

EVENTS, ADVANCES, AND NEWS

■ BY JEFF ECKERT

ADVANCED TECHNOLOGY

INTERMITTENT SHORT DETECTOR USES PULSE TECHNOLOGY

■ Mike Dinallo (right) and Larry Schneider (left) prepare to employ the PASD diagnostic on a wiring bundle in the cockpit of a retired Boeing 737. Photo by Randy Montoya.

It may not sound all that complicated, but the folks at Sandia National Laboratories (**www.sandia.gov**) have spent decades developing the concept of pulsed power, and in particular the version generated by its Z machine, which sends short bursts of trillions of watts down conduits with diameters the size of your average horse. But a recent adaptation, known as the Pulsed Arrested Spark Discharge (PASD) operates on a much smaller and more mundane level and, in fact, will soon be available from a commercial licensee, Astronics Advanced Electronic Systems (**www.astronicsaes.com**). The basic concept is to introduce a high-voltage, low-power pulse into an aircraft wiring harness (up to 40 wires at a time) for just a nanosecond to detect intermittent shorts that can result in expensive downtime and even crashes. Because the voltage is higher than that normally used in airplanes, the electrical pulse will jump from the smallest wiring insulation fault (which to ordinary instrumentation seems undamaged) to either the bulkhead or another nearby damaged wire. That spark, much like an electrostatic discharge event, in effect lights up the invisibly damaged spot. The amount of time it takes for the current to return to its source is analyzed by the automated test set to measure, within inches, how far the break is from the test entry point. These breaks can be difficult to locate visually, because the wiring may have tiny insulation breaks the size of a pinhole or a cut from a razor blade, and traditional wire-test systems have great difficulty finding these faults, as well.

To enhance the tester's fault-locating ability, Astronics has developed a method to superimpose the PASD pulse on a DC current. The DC current provides support for the high-voltage pulse, which then can detect breaches as far as 100 feet from its starting point — even those occurring on branched wire harnesses. The distance to a fault is computable, regardless of changes in impedance produced by the wiring as it reacts to the PASD pulse at various voltage levels. Other possible uses envisioned for PASD are to perform inexpensive tests on the wiring of passenger cars and new homes. Military tanks and the hard-to-reach wiring behind the steel bulkheads of submarines are also possible candidates.

NEW X-RAY DELIVERY METHOD FOR RADIATION THERAPY

Researchers at the US Department of Energy's Brookhaven National Laboratory have announced that, following many years of investigation, improvements in a heretofore experimental form of radiation therapy appear to make the technique more effective and eventually may allow its use in hospitals. Results on the improved method, which was tested on rats, was published online by the Proceedings of the National Academy of Sciences and, at least as of this writing, is viewable at **www.pnas.org/cgi/doi/10.1073/pnas.0603567103**. The technique, developed in cooperation with Stony Brook University, the IRCCS NEUROMED Medical Center in Italy, and Georgetown University, is known as microbeam radiation therapy (MRT). Originally, it simply used a high-intensity synchrotron x-ray source to produce parallel arrays of very thin (25 to 90 m) planar x-ray beams rather than the solid, broad beams used in conventional radiation treatment. Studies have shown that MRT can control malignant tumors in animals with high radiation doses while subjecting adjacent normal tissue to little collateral damage. A drawback, however, is that only certain synchrotrons can generate these very thin beams at adequate intensity, and such facilities are available at only a few research centers around the world. However, the paper discusses how thicker microbeams — which can be generated by more common x-ray tubes of very high current and voltage — may work, as well. Plus, the authors describe a way to interlace the beams to increase their killing potential while retaining the technique's hallmark feature of sparing healthy tissue outside the target.

In an experiment, they first exposed the spinal cords and brains of healthy rats to thicker (0.27 to 0.68 mm) microbeams at high doses of radiation and monitored the animals for signs of tissue damage. After seven months, animals exposed to beams as thick as 0.68 mm showed no or little damage to the nervous

system. Next, the scientists demonstrated the ability to interlace two parallel arrays of the thicker microbeams at a 90-degree angle to form a solid beam at a small target volume in the rats' brains and then measured the effects of varying doses of radiation on the targeted and surrounding tissue using magnetic resonance imaging (MRI) scans. For interlacing, the gaps between the beams in each array were chosen to be the same as the thickness of each beam, so the beams from one array would fill the gaps in the other to produce the equivalent of an unsegmented beam in the target volume only. The MRI scans showed that, at a particular dose of radiation, the new configuration could produce major damage to the target volume, but virtually no damage beyond the target range. Neither the original nor the improved MRT technique has been tested in humans, but is appears to have a great deal of promise.

COMPUTERS AND NETWORKING

COMPUTER SECURITY THREATS DOUBLE

If you have a feeling that threats to computer security are getting worse, you're right. McAfee, Inc. (**www.mcafee.com**), has announced that it now has 200,000 definitions in its threat database, which contained only half that number in September 2004. It took 18 years for the database to reach 100,000, so we're looking at a 60 percent decrease in the amount of time it took to double in size. Furthermore, McAfee predicts that the number will reach 400,000 in less than two more years. According to the company, while bots are still the leading cause of this dramatic growth, exploits and downloaders are a close second. Email threats, which made up a large percentage of the number of threats in 2004, saw smaller growth over the last two years as compared to other categories of "malware." The company warns that to stay protected, both enterprises and consumers must constantly stay updated with the latest virus signature files, install the latest patches, and implement a multilayered approach to detecting and blocking attacks. For more information on this subject — including a specific breakdown of growth rates by type of threat — or to learn more about cutting-edge security research and opinions, you can visit the McAfee Avert Labs Security Blog at **www.avertlabs.com/research/blog/?p=49**.

INTERNET FAX SERVICE ENHANCED

If you're ready to dump the fax machine and start sending all your faxes via the Internet, you might be interested in a service called MyFax® (**www.myfax.com**), which is actually provided by Protus IP Solutions, Inc. For $10 per month, you can send and receive faxes using existing email accounts, and the base rate includes 100 sent and 200 received pages. Inbound faxes are stored for 31 days on a secure server. The latest news from the company is that MyFax now is integrated with Microsoft® Office 2003 and Lotus Notes®, so you can send a fax directly from Word®, Excel®, or PowerPoint® using a single button with no need to open Outlook® first. The upgraded service also provides increased file support for outbound faxing, including Visio, Snapshot, and MHT files.

LOWER PRICES FOR HARD DRIVES

Continuing the traditional industry trend toward more storage for less money, Newer Technology, Inc. (NewerTech), has announced price cuts on its full line of miniStack V2 FireWire/USB 2.0 combination storage devices. Prices now range from $124.99 for the 80 GB unit to $559 for a 750 GB model. A nice mid-range selection is the 500 GB miniStack, which has dropped in price from $419.99 to $299. All models 160 GB or higher feature the latest Seagate Barracuda 7200.9 and 7200.10 mechanisms, which carry a five-year warranty. The 80 GB model is based on Hitachi hard disk mechanisms with a three-year warranty. The overall package is warranted by NewerTech for two years and takes up only 6.5 by 6.5 by 1.5 inches. It works with any Mac, Intel Mac, PC, or Linux computer with an available FireWire or USB port. For details, visit **www.newertech.com**.

■ The miniStack V2, showing rear and side ports.

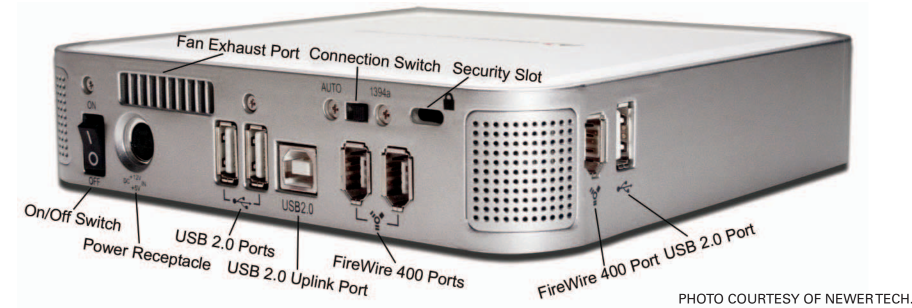


PHOTO COURTESY OF NEWERTECH.

CIRCUITS AND DEVICES

IPOD® TO CAR STEREO CONNECTION

Assuming that the piece of junk you're driving around is old enough to sport a cassette player instead of a CD changer, Belkin has a deal for you. The TuneDeck™ for iPod nano (model F8Z064) is a complete in-car device

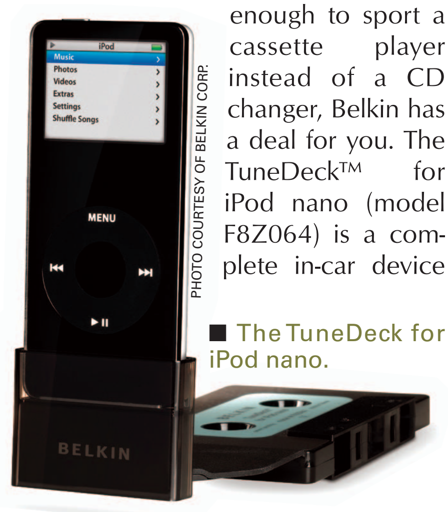


PHOTO COURTESY OF BELKIN CORP.

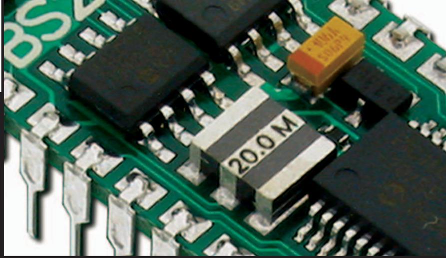
■ The TuneDeck for iPod nano.

that allows you, by plugging it into the cassette deck, to port the iPod's output into your car stereo. It also powers and charges the iPod through the included mobile power cord, holds it in place with a custom cradle, and offers a swivel base so you can position the screen conveniently. The unit costs $49.99 and will be available in the USA by the time this article goes to print, with distribution to follow shortly afterwards in Asia, Europe, and Australia. **NV**

INDUSTRY AND THE PROFESSION

WIND-TO-HYDROGEN PROJECT ANNOUNCED

The US Department of Energy's National Renewable Energy Laboratory (NREL) and Xcel Energy (**www.xcelenergy.com**) recently signed a cooperative agreement for an innovative "wind to hydrogen" research, development, and demonstration project. Researchers will analyze and compare hydrogen production from wind power and the electric grid. The hydrogen will be produced through electrolysis. The agreement supports the President's Hydrogen Fuel Initiative, which seeks to develop the hydrogen, fuel cell, and infrastructure technologies needed to make it practical for Americans to use fuel cell vehicles by 2020. The wind-electrolysis system will be at NREL's National Wind Technology Center, where hydrogen will be produced, compressed, and stored to be used as a vehicle fuel or to generate electricity. The project will compare electrolyzer technologies and researchers will examine issues related to system efficiency, integration, compression, storage, cost, and the use of a mixture of hydrogen and natural gas. According to an Xcel representative, "Xcel Energy is the nation's leading wind energy utility with 2,300 megawatts of capacity planned for our system by the end of next year," said Frank Novachek, director of Corporate Planning at Xcel Energy. "This project will help us explore how we can leverage clean, renewable, yet intermittent power sources like wind into a more valuable resource for our utility customers." Xcel Energy plans to invest more than $1.25 million in the project, and NREL and the Department of Energy plan to invest approximately $750,000.



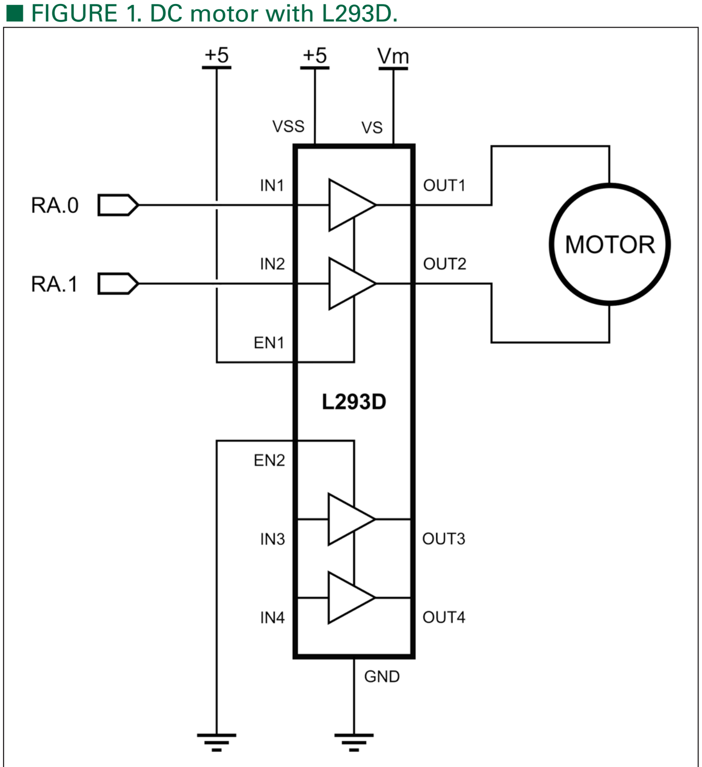
■ BY JON WILLIAMS

SX/B TURNS SWEET 16

THERE ARE THOSE — THE PESSIMISTS AMONG US — that will insist that you can't get anything worthwhile for nothing; everything has a price. Not so with SX/B. While it may not compete with big, "professional" compilers, in the right hands (i.e., yours) and with a few tricks, SX/B is quite capable and costs absolutely zero dollars. And with the cost of the SX-Key programming tool and SX Proto Boards so low these days, it's really hard to ignore the SX micro as a viable solution to many design problems.

Truth be told, it's easy for me to be a fan of SX/B because I was part of the team that developed it. Still, those of you who know me understand that I'm a very practical guy; I don't have a lot of time to fool around and when I need something, I need it, and I need it to work. Since leaving Parallax for new adventures, I have, in fact, continued to use SX/B — I recently designed a camera controller using an SX28 that was programmed entirely in SX/B 1.5x (no assembly language required). My point is that SX/B wasn't developed simply for the sake of doing it; SX/B was developed to provide a practical, no-cost tool for SX developers.

■ FIGURE 1. DC motor with L293D.



If you've never tried the SX, perhaps this article will encourage you to do so. It really is hard to beat the cost of entry: the SX-Key (ICP programming tool with full debugging capability) is only $79, the SX-Blitz (programming only, great for students) is an incredible bargain at only $29, and the fully-populated (power supply, SX chip, connectors) SX48 Proto Board is only $10! Yes, ten bucks. Using the Blitz, the SX48 Proto Board, a serial cable and a 12 VDC power supply, you could get into SX programming for about $50 — that's really not a bad deal for all the horsepower delivered by the SX.

SX/B 1.5X

The big news with version 1.5x is, of course, the addition of Word (16-bit) variables. This is especially good news for BS2 users wanting to port prototype projects to the SX for high-volume production. As in PBASIC, we declare a 16-bit variable in SX/B as type Word:

```
tmpW1           VAR      Word
```

When we look "under the hood" of SX/B (use Ctrl+L to compile and view the listing), we'll see that the definition above is actually composed of two bytes:

```
tmpW1           EQU      0x0D
tmpW1_LSB       EQU      tmpW1
tmpW1_MSB       EQU      tmpW1+1
```

Note that the value is stored "Little Endian" (low byte first) and that in addition to the name we declare, the compiler creates definitions with the suffixes _LSB and _MSB; these byte variables can be used just as we would use the *tmpW1.LOWBTYE* and *tmpW1.HIGHBYTE* notation in PBASIC.

Word variables can be used exactly as we'd expect — and even in a few ways that we might not consider at the

start. The only caveat is this: Due to the limit of internal variables used by SX/B operations, we cannot multiply a word variable by itself and return the result to that same variable. The following line of code will generate a compiler error:

```
  tmpW1 = tmpW1 * tmpW1
```

We can use the other operators here (+, -, /), just not operators that involve multiplication (*, */, and **).

Most of the SX/B instructions have been upgraded to work with Word variables, and a new variant of the **DATA** directive, called **WDATA** lets us store Word values for use with **READ**. The use of 16-bit values extends to I/O ports, as well. In SX/B 1.5x, there are three 16-bit pseudo ports: RBC, RCD, and RDE; the last two only apply to the SX45/52.

Here's a simple demo that shows how we can use the RBC port on an SX28:

```
Start:
  TRIS_B = %00000000
  TRIS_C = %00000000

  RBC = %00000000_00000001

Main:
  DO
    DELAY 75
    RBC = RBC << 1
  LOOP UNTIL RBC = %10000000_00000000
  DO
    DELAY 75
    RBC = RBC >> 1
  LOOP UNTIL RBC = %00000000_00000001
  GOTO Main
```

The program starts by making the RB and RC ports outputs — we have to do it this way because there is no TRIS_RBC port. The RBC port gets initialized and falls into a loop that simply ping-pongs the lit LED back and forth. Note the use of the underscore character in the comparison statement to make visualization of the 16-bit value easier.

Since this program uses a subroutine called ***DELAY***, and we might want to do delays with 16-bit values, let's look at the construction of subroutines in SX/B 1.5x.

For ***DELAY***, we'll use the following declaration:

```
DELAY           SUB     1, 2
```

This will let us pass a one- or two-byte value to ***DELAY***. Here's the actual subroutine code:

```
' Use: DELAY ms
' — 'ms' is delay in milliseconds, 0 - 65535

DELAY:
  IF __PARAMCNT = 1 THEN
    tmpW1 = __PARAM1
  ELSE
    tmpW1 = __WPARAM12
  ENDIF
  PAUSE tmpW1
  RETURN
```

The construct of this subroutine is useful in many other situations as it allows us to pass bytes or words to the same subroutine. When we pass a byte, *__PARAMCNT* (internal variable) will be set to one by the compiler and the parameter is passed in *__PARAM1*. When we pass a word, *__PARAMCNT* will be set to two and the value passed in ***__WPARAM12***. Of course, we'll use a word-sized variable in the subroutine so that we can handle anything passed to it.

FUNCTIONAL SUBROUTINES

With the addition of 16-bit variables, a mechanism needs to be developed that would enable Word values to be returned from a subroutine; this is accomplished with the **FUNC** definition. This lets us define a function that can return up to four bytes (two words).

FUNC differs from **SUB** in that we will first specify how many bytes are to be returned, then the minimum and maximum parameter count used by the subroutine code for the function. Let's say that we wanted a function that would return a 32-bit product from two numbers — can we do it? Yes, of course! We don't have 32-bit values in SX/B, so we have to handle the words separately. Let's start with the function definition:

```
MULT32          FUNC    4, 2, 4
```

This definition says that the function, ***MULT32***, will return four bytes, and that the caller must pass between two and four bytes to it. This means that we could return the product of two bytes, a word and a byte, or two words. Note that the second option — multiply a word and a byte — can create some trickery for our subroutine construction, so we must make a decision about the order that values are passed. Let's decide that we will pass the word value first, then the byte. Here's the code for that function:

```
MULT32:
  IF __PARAMCNT = 2 THEN
    tmpW1 = __PARAM1
    tmpW2 = __PARAM2
  ENDIF
  IF __PARAMCNT = 3 THEN
    tmpW1 = __WPARAM12
    tmpW2 = __PARAM3
  ENDIF
  IF __PARAMCNT = 4 THEN
    tmpW1 = __WPARAM12
    tmpW2 = __WPARAM34
  ENDIF
  tmpW3 = tmpW1 ** tmpW2
  tmpW2 = tmpW1 * tmpW2
  RETURN tmpW2, tmpW3
```

As with the ***DELAY*** routine we did earlier, this code uses the *__PARAMCNT* variable to determine what is being passed and how to collect the parameters from the caller. The second choice, when *__PARAMCNT* is three, assumed that the first value passed is the word and the second is the byte. With the parameters collected, the rest is easy; the **

operator (new in SX/B 1.5x, and */ has been added, as well) returns the upper 16 bits from a 16-bit x 16-bit multiplication. The * operator will return the lower 16 bits of the product.

Note how the 32-bit value is returned to the caller as two words, separated by a comma, low-word first. So how do we collect this 32-bit value? Let's start with variables to hold it:

```
result          VAR     Word
resultHi        VAR     Word
```

And here's how we can use the function in a program.

```
  result = MULT32 $FFFF, $0100
  resultHi = __PARAM3, __PARAM4
  BREAK
```

The first part is obvious, I'm sure; we call the function and assign it to result. But this only gets us the lower 16 bits. To get the upper 16 bits, we have to grab them ourselves. The high word from the function will be returned in *__PARAM3* (LSB) and *__PARAM4* (MSB). This also demonstrates how to move two bytes into a word with just one line of code.

There is a method for collecting all four bytes from this function without the second line of code above — but we must use an array as the target variable. So, we could do this:

```
bigVal          VAR     Byte(4)
result          VAR     bigVal(0)
resultHi        VAR     bigVal(2)
```

And now we just need one line of code:

```
  bigVal = MULT32 $1234, $10
```

One of the interesting things about the SX-Key tool is that it will let us view 32-bit values in the Debug window. We can set up a **WATCH** declaration like this:

```
WATCH result, 32, UHEX
```

If we run the program in Debug mode with a **BREAK** instruction after the function call, we'll see the 32-bit result.

PIN DOWN YOUR I/O

One of the latest updates to SX/B is the **PIN** definition that became available in version 1.51. In the past, we might define an I/O pin like this:

```
Led             VAR     RC.0
```

Now we can do this:

```
Led             PIN     RC.0  OUTPUT
```

What's the advantage? Well, the compiler will automatically insert startup code that makes the pin an output, so we don't have to worry about anything beyond the declaration. That way, we can write directly to the pin knowing that the appropriate TRIS register has been set up correctly.

In a lot of my older programs, I would enable the SX pull-ups on any unused pin to minimize current draw. It's even easier now. Let's say that we have just the one LED as above. By using the following declarations, we don't have to worry about TRIS or PLP register settings in our code, which lets us focus solely on the application. Note how **PIN** works with groups and individual I/O pins.

```
UnusedA         PIN     RA    INPUT  PULLUP
UnusedB         PIN     RB    INPUT  PULLUP
UnusedC         PIN     RC    INPUT  PULLUP

Led             PIN     RC.0  OUTPUT NOPULLUP
```

The final declaration overrides the definition for RC.0 from the group above; this way, we can define the unused pins as a group instead of one at a time.

It's important to understand that generation of **PIN** start-up code is enabled even when the **NOSTARTUP** option for the **PROGRAM** directive is specified. The available options for **PIN** are INPUT, OUTPUT, PULLUP, NOPULLUP, TTL, CMOS, and SCHMITT — and when multiple options are used, they are space-delimited.

INTERRUPTS WITHOUT IRRITATION

Before I get too far into this section, let me start by saying that interrupts are always tricky but that SX/B 1.5x does make them a bit easier to cope with. With SX/B 1.5x, we can simply specify how frequently (in interrupts per second) that the ISR should run and the compiler will take care of the rest, setting the OPTION register and the **RETURNINT** value automatically. Let's start with a very simple example:

```
' ------------------------
  INTERRUPT NOPRESERVE 1000
' ------------------------

ISR_Start:
  INC timer
  IF timer = Cycles THEN
    TOGGLE Led
    timer = 0
  ENDIF
  RETURNINT
```

The purpose of this code is to toggle the state of an LED every *N* milliseconds, defined by the program constant called **Cycles**. Note that the end of the **INTERRUPT** declaration line specifies 1000 — this will cause the program to set up the interrupt such that it runs once every millisecond. If we specify an ISR rate that that won't work with the **FREQ** directive, the compiler will complain of an invalid parameter.

This is interesting, but we may not want to blink the LED with a 50% duty cycle. Here's an easy update that allows us to specify the on- and off-time for the LED.

```
' -------------------------
  INTERRUPT NOPRESERVE 1000
' -------------------------

ISR_Start:
  INC timer
  IF Led = IsOn THEN
    IF timer = OnTime THEN
      Led = IsOff
      timer = 0
    ENDIF
  ELSE
    IF timer = OffTime THEN
      Led = IsOn
      timer = 0
    ENDIF
  ENDIF
  RETURNINT
```

You've probably noticed that these programs use the **NOPRESERVE** option in the **INTERRUPT** declaration and may be wondering why and when to use this option. The reason why is that it will reduce the amount of code in the ISR. When can we use this option? We can use **NOPRESERVE** when none of the SX/B internal variables are being used in the ISR. This can be determined by using Ctrl+L to compile the program and show the assembly listing; if none of the internal variables (*__PARAM1 – __PARAM4*) are being used, then **NOPRESERVE** can and should be used.

Before we wrap up this section, let's take the second version of the LED blinker and use it to drive a motor. Remember the L293D that we used in the stepper project last month? Well, it's a push-pull driver so we can use two of its channels to drive a small DC motor and have control over speed and direction with just two I/O pins.

One pin will be pulse width modulated by the ISR to provide speed control. The other pin will determine the direction that the motor spins. We could add control of the L293D enable pin, as well, but this program assumes that it is tied high.

Let's look at the ISR first:

```
' ---------------------------
  INTERRUPT NOPRESERVE 10_000
' ---------------------------

ISR_Start:
  INC phase
  IF phase > 100 THEN
    phase = 0
    IF m1Speed > 0 THEN
      M1Ctrl = IsOn
    ENDIF
  ELSE
    IF phase > m1Speed THEN
      M1Ctrl = IsOff
    ENDIF
  ENDIF
  RETURNINT
```

Looks pretty simple, doesn't it? In fact, it is. The code starts by incrementing a variable called ***phase*** — this tracks where we are — 0 to 100% — in the PWM cycle for the motor. When that value exceeds 100, we reset everything by clearing the phase counter and turning the motor control output on (if the speed is not set to zero). During the rest of the cycle, we compare the phase value to the speed of the motor; as soon as ***phase*** exceeds the motor speed, the motor is shut off. The behavior of this code lets us specify the motor speed in percentage.

The ISR runs the motor, but we need a subroutine to set the speed and direction when we need a change.

```
SET_MOTOR:
  tmpB1 = __PARAM1
  tmpB2 = __PARAM2
  m1Speed = tmpB1 MAX 100
  IF tmpB2 = Fwd THEN
    M1Dir = Fwd
  ELSE
    m1Speed = 100 - m1Speed
    M1Dir = Rev
  ENDIF
  RETURN
```

This code, too, is very straightforward. After collecting the parameters, the speed is set, limiting its value to 100. Then the direction pin (second motor control output) is set. Here's where we need to make an adjustment when reverse direction is specified. The ISR always makes the motor control pin high during the "on" phase of the motor. This is fine when the direction is set to forward and the direction pin is low, but when the direction pin is high (for reverse), what was the "on" time of the motor actually becomes the "off" time. Don't worry, the solution is simple. All we have to do is "invert" the reverse speed value by subtracting it from 100.

From my point-of-view, motor PWM control is a bit of black art. Luckily, the code is really easy to update. I found that setting my ISR rate to 10,000 (which works out to a 100 Hz PWM frequency) worked best for the motor I was using. If this setting was too high, the motor wouldn't move at low speeds; if it was too low, the movement was very choppy at low speeds. You may need to experiment with your motor.

Finally, we must remember that when the ISR is enabled as in the previous examples, it "steals" time from the rest of our program and will affect time-sensitive instructions like **PAUSE** and **PAUSEUS** (they get longer), and **SERIN** and **SEROUT** may not work at all. Advanced programmers will appreciate the *Effective-Hertz* parameter of the **FREQ** directive in SX/B 1.5x. If the ISR code runs a fixed period, then we can determine the "effective" clock frequency when the ISR is active and allow the compiler to generate code that will operate as expected.

SX/B WITH STYLE

In the SX/B 1.5x help file, you'll find a section called "The Elements of SX/B Style." This was, of course, adapted from "The Elements of PBASIC Style" that appears on the Parallax website and in the PBASIC help file.

The key to success with SX/B, I believe, is using subroutines and functions properly. If you do this, for example:

```
SERIN char1
SERIN char2
SERIN char3
```

You'll chew up a whole bunch of code space as each **SERIN** instruction is expanded to the assembly code required for that function — there is no automatic optimization by the compiler. Optimization, then, is the responsibility of the programmer, and the easiest way to do it is put "big" instructions into subroutines and functions.

What's a "big" instruction? It is any instruction that expands to more than a few lines of assembly code; most of the instructions that have any sort of timing element will fall into this category, things like **SEROUT**, **SERIN**, **PAUSE**, etc.

One final note on **SUB** and **FUNC** declarations: When the subroutine code does not require any parameters, use 0 in the declaration — like this:

```
RX_BYTE      FUNC     1, 0
```

This will save a bit of generated code — just a bit — but every little bit counts with small micros, right?

SX/B 1.5x has a couple more tricks up its sleeve you'll like: **COUNT** and **COMPARE** instructions (ala BS2), **TIMER1/TIMER2** instructions that simplify the use of the SX48/52 multi-purpose timers, and an option I particularly like is the clock speed multiplier for **SHIFTIN** and **SHIFTOUT**; this lets us connect to synchronous devices at (or very near) their maximum clock speed.

It's your turn now; if you're already using SX/B, make sure you download the latest version (it's free!), and if you're not using the SX, why not? When one considers the cost of entry, a free compiler like SX/B, and the horsepower the chip can deliver ... in my book, it's a great value and should be part of your arsenal. Give it a try — you'll be glad you did.

So good-bye for now and, until next time, Happy Stamping — SX/B style! **NV**

RESOURCES

JON WILLIAMS
jwilliams@efx-tek.com

PARALLAX, INC.
www.parallax.com

Q&A

WHAT'S UP:

Something I've been wanting to do for a long time: Modify a laser pointer.

- ✔ 50 ways to debounce an input.
- ✔ Laser pointer transmitter/receiver pair.
- ✔ Dolby 7.1 defined.
- ✔ Automatic airport lights.

■ WITH TJ BYERS

In this column, I answer questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist.

Feel free to participate with your questions, comments, or suggestions.

You can reach me at: **TJBYERS@aol.com**

FOLLOW THE DEBOUNCING BALL

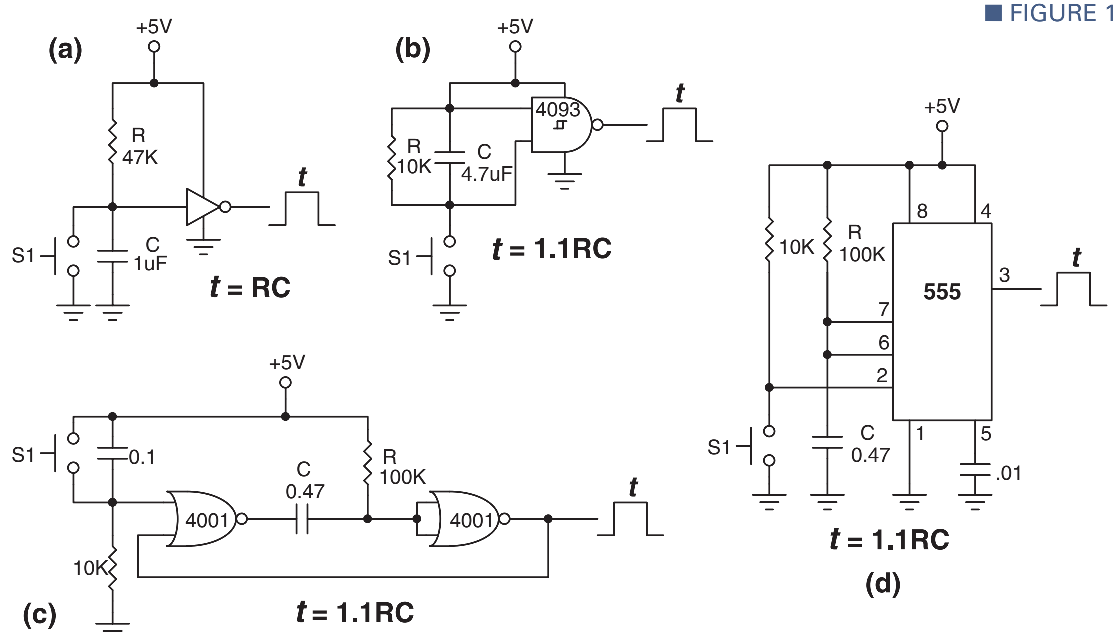
Q I have an interest in mechanical switch contact bounce. I have been very successful in accounting for contact bounce using software, but have never tried to eliminate it in hardware. Last month, I read a statement in a book about electronic circuits that said switch contact bounce could "easily be eliminated by the fitting of a small capacitor" into the circuit. No details were given. I experimented for hours at the oscilloscope using a variety of capacitors, but never came close to eliminating the contact bounce. Are you familiar with using a capacitor to eliminate bounce? If so, how do you determine the value of the capacitor, and how is it placed in relation to the switch?

— Judy May W1ORO
Union, KY

A All mechanical switches and relays have leaf-like contacts with mass and flexibility. Because of these properties, the contacts will "bounce" upon closure for a period of up to 20 milliseconds (ms) before coming to rest and providing an unbroken contact. (Contact bounce can also occur when the switch opens, to a lesser extent.) During this time, the switch can send many pulses to the input of a microcontroller or logic gate, creating false triggering.

There are several methods to eliminate contact bounce — including the software routines you have mastered. At the hardware level, contact bounce is eliminated using RC time constants. When a resistor and a capacitor are paired, they form a time-dependent network, as shown in Figure 1(a). When switch S1 is pressed, it shorts out the capacitor and forces the input of the logic gate to ground,

■ FIGURE 1



Switch Debouncer

which generates an output voltage. When the push button is released, the capacitor charges through the 47K resistor at a rate of *t* = RC. When the voltage across the capacitor reaches about 3.0 volts, the gate flips logic and the output goes low.

Now how does this help us with contact bounce? If we select the time constant of the RC network so that it takes more than 20 ms to reach 3.0 volts, any voltage excursions of the contact bounce will have their tops lopped off like a lawnmower over grass. The typical debounce time is anywhere between 50 ms and 0.1 seconds. If the resistor is 47K and the debounce time is 50 ms, then the capacitor is 1 µF (C = *t* / R).

While this design works well for driving most CMOS devices, TTL and low-voltage logic chips require a cleaner pulse. Often a Schmitt trigger (Figure 1(b)) is used in place of an inverter. Unlike a standard logic gate — which has its high and low trigger voltages at three and two volts, respectively — a Schmitt trigger has a greater hysteresis with trip points at 3.3 and 1.8 volts. This wider voltage band lets the Schmitt trigger clean up contact bounce that would otherwise pass through a standard gate. Notice the inverse arrangement of R and C — they are in parallel instead of serial. In this arrangement, C charges when S1 is closed and discharges through R when it opens. When both NAND inputs are high, the output goes low.

For really dirty switches, you need to bring out the big guns — one-shot monostable multivibrators. Monostable multivibrators are sometimes called pulse stretchers because once triggered, the output remains high until the one-shot times out. No amount of hammering on the input can disturb this output in its duties. Figure 1(c) and (d) show two monostable designs. Circuit (c) uses a pair of NOR gates to form the one-shot. Circuit (d) isyour classic 555 monostable multivibrator.

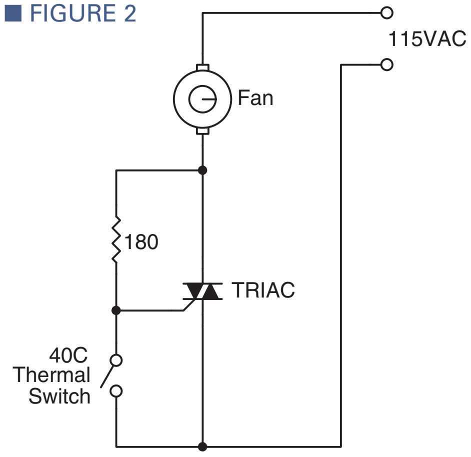
A/C FOR A/V BUILD-IN

Q I have a multi-media cabinet with a small 120 VAC muffin fan that I would like to control with a temp switch: on at 85°F, off at 80°F. You mentioned a stereo fan cooler in the June '02 issue, but didn't show a schematic. I need a schematic.

— Mike

A Actually, there was a schematic — but it's not one that you can use because it was a variable speed controller for a 12-volt fan. What you need is a thermal switch, like the R2004025 switch from Cantherm and available from Digi-Key (800-344-4539; **www.digikey.com**). See Figure 2.

FIGURE 2



Multi-Media Cooling Fan

This switch is normally closed (NC) and opens when the temperature inside the cabinet exceeds 104°F (40°C). (I know you wanted 85°F, but 104°F is more practical.) Basically, if the cabinet temperature is below that of the bi-metal switch, the contacts remain closed and prevent the triac and fan from turning on. When the temperature inside the cabinet exceeds the switch's limit, the fan turns on and stays on until the temperature drops significantly.

Why a triac instead of powering the fan from the switch itself? Control. This way, you can up the switched current by simply replacing or paralleling the triac to operate any number or type of fans using the same switch.

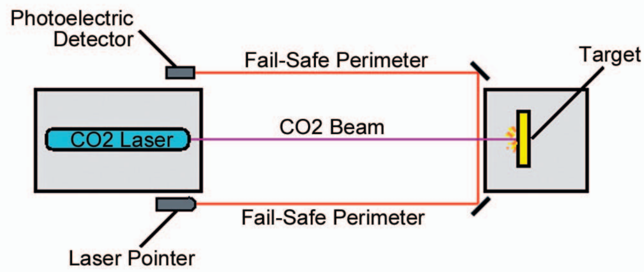
PERIMETER SECURITY

Q Having just finished co-nstruction of a CO_2 laser, I would now like to know how I could build a simple radio transmitter with a momentary contact switch that would allow me to trip off a 110 VAC switch at 10 or 20 feet away for emergency purposes. When doing experiments, sometimes things go wrong that start a fire and I need to be able to turn off the laser FIRST and THEN put out the fire. I don't want to inadvertently walk into the invisible beam while solving the emergency.

— Bob Slusher

A Yipes! I wouldn't rely on a radio signal to prevent a laser beam from sawing me off at the knees. Instead, I would ring the laser experiment with another laser beam — actually a photoelectric beam that would switch off the 110 VAC if broken.

I would begin with a laser pointer — the kind you can buy for a couple of bucks. Using adjustable mirrors, ring the experiment area with the pointer beam — whose red dot you can see when it strikes a hard surface, like a sheet of white paper — and direct it to a photo detector. See Figure 3.



Do Not Enter Fail-Safe (Top View)

FIGURE 3

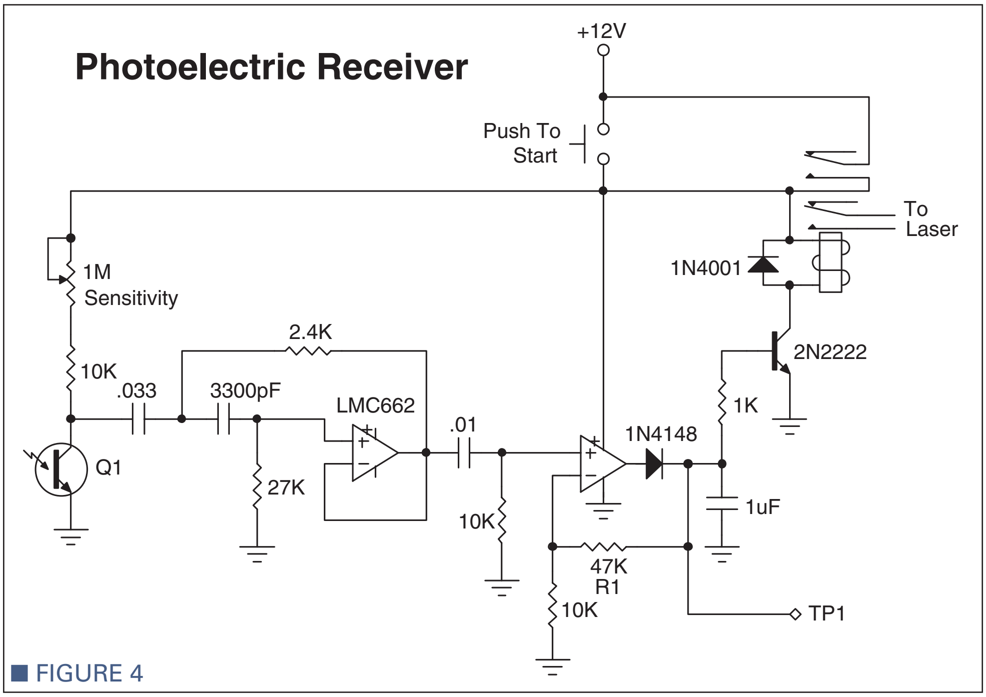


FIGURE 4

Let's start with the photo detector, for which I suggest a phototransistor. They're cheap, readily available, and sensitive to red light. The collector of a phototransistor outputs a voltage that's proportional to the light falling on its lens. This is all well and good if you want to monitor the difference between night and day, but our concern is the laser pointer beam. Of course, we could provide lots of optical "tunnels" to shield the sensor from ambient light, but I have a better solution.

Modulate the laser beam so that the phototransistor responds only to a modulated signal and ignores ambient and fluorescent light. This is done using Figure 4. The phototransistor (Q1) is AC coupled to a high-pass Chebychev filter, which removes the DC component (ambient light) and 60 Hz interference from fluorescent lamps. The filtered signal is input to a precision rectifier with a gain of about 6x. With the values shown, a 200 mV, 2 kHz output signal from Q1 is enough to engage the 12-volt relay. If you need more amplification, simply increase the value of R1; e.g., 100K boosts the gain to 11.

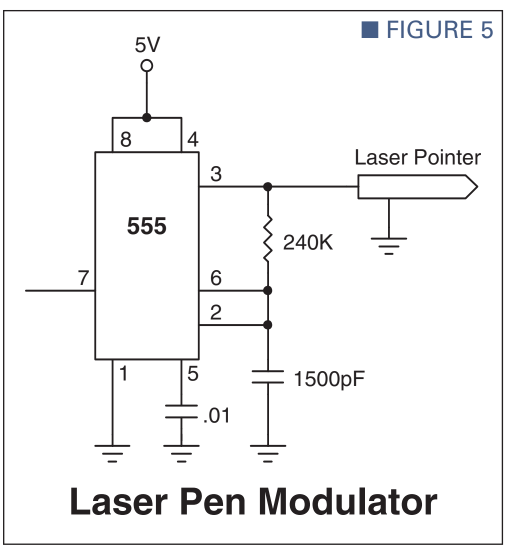


FIGURE 5

The transmitter is a modified laser pointer. The key chain type is cheapest and easiest to modify, so I'll discuss its modification. *Warning: Never look a laser beam straight in the eye. Use white cardboard cards to adjust the mirrors. And don't be tempted to remove the laser diode from the barrel. It's used as a heatsink to cool the diode.*

That said, remove the batteries and locate the spring that connects to the button of the battery. With a pair of long-nose pliers, carefully unwind the spring so that it's accessible from outside the barrel and short the ON button using whatever means works for you. (I have used shrink tubing to successfully close the switch on some models.) Slip a length of shrink tubing over the extended spring to keep it from shorting to the case. This is your sender, which now has to be modulated. I use a 555 because — when configured as in Figure 5 — it has the fewest parts for a square-wave oscillator and can readily supply the 40 mA of current the laser diode requires without having to use an external transistor.

To use the fail-safe, you first align the mirrors so that the red dot falls squarely on the phototransistor. With the CO_2 laser disconnected, hold down the Push To Start button and adjust the Sensitivity potentiometer for maximum DC voltage on TP1. If the relay won't pull in, increase the value of R1. (In extreme cases where the ambient light floods the phototransistor, you may need to shield the sensor in a two-inch length of 1/2" PVC pipe.)

Release the Push To Set button, turn off the laser pointer, and connect the CO_2 laser. To start your experiments, energize the laser pointer, press the start button — and have fun frying. If the perimeter is breached, the beam will be interrupted and the relay will be disengaged — until it's reset with the Start button.

IS EIGHT ENOUGH?

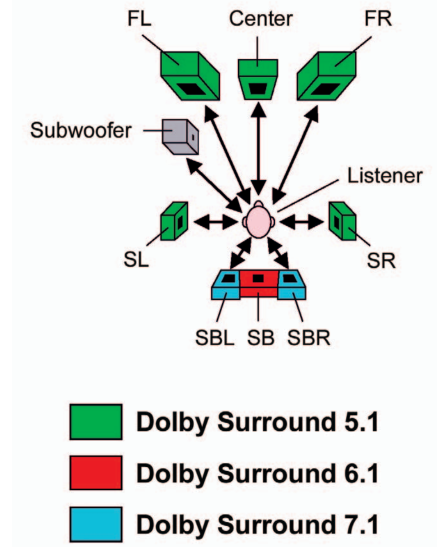
Q I'm a service technician at a TV/Stereo repair shop in New Jersey. We're looking for a test disc (could be audio with video, or just audio alone) for an SACD/DVD Audio/Video Player that will output 7.1 Dolby Surround. That would mean eight channels, each with its own signal and identifier. Please let me know of any source. Also, I'm kinda new to surround sound, and an explanation of what the designations mean would be nice.

— John Agugliaro, CET

A Surround sound comes in many versions, with 5.1, 6.1, and 7.1 being the most popular. The first number is the number of "audio" speakers, with the x.1 designation referring to the sole subwoofer — with a frequency range extending from 80 Hz down to 10 Hz. Typically, surround sound is an evolution of 1950's stereo and 1970's quad channels to create the effect of theater sound in a small space. This is done by recording six discrete channels of audio on the recording media (Figure 6) — 5.1 Surround Sound. From this, all the other versions have been derived. That is, there are only six audio channels — nothing more.

In 6.1 Surround Sound, an extra rear speaker is added. This is a mono channel created by adding the two rear left and rear right channels together and placing it behind your back where the sound resembles an echo of the Center speaker. In 7.1 Surround Sound, the back speaker is split into two. Again, the sound is derived from the two rear side channels. Sometimes the two sound sources are a combination of L-R and L+R (not unlike FM stereo) to create the background sound. Sometimes the sounds are simply a duplication of the back speakers spaced accordingly, or 6.1 decoding fed into both. Neither the 6.1 or 7.1 formats have separated channels. They are an algorithm of the original five 5.1 Surround Sound.

Given these parameters, I often question who benefits from these new formats. Unless you have a video room the size of a pool hall, most readers won't hear the difference and should not be bilked into paying for something they can't use. Oh, about your test disc, try these people: **http://ferradoylegroup.com/dolby/dol302.html**. I ordered one of these discs, and found it well worth the $5 asking price.



■ FIGURE 6

PROGRAMMERS WANTED

Q I'm looking for a circuit with which to build a six-digit clock, just as I used to do with National Semiconductor's MM5314, the T-I MM5375, and the Mostek MK50250 (my favorite). Unfortunately, all the above are obsolete, so what I want is a number for their replacement — not those excuses you make for PICs! A chip that delivers a multiplexed six-digit output and uses the line frequency for timing — which is usually more accurate than crystal time base. I also doubt that clock radios use the esoteric circuits you described. There must be a source for simple multiplexed clock chips. What chips are used in the clock radios you can find in many stores, or in the kits sold by Jameco, Ramsey, and others?

A This is a very common request — and one I have a hard time filling. The following two places say they have at least one of these items in stock, and while they don't require a minimum order, expect to pay premium prices.

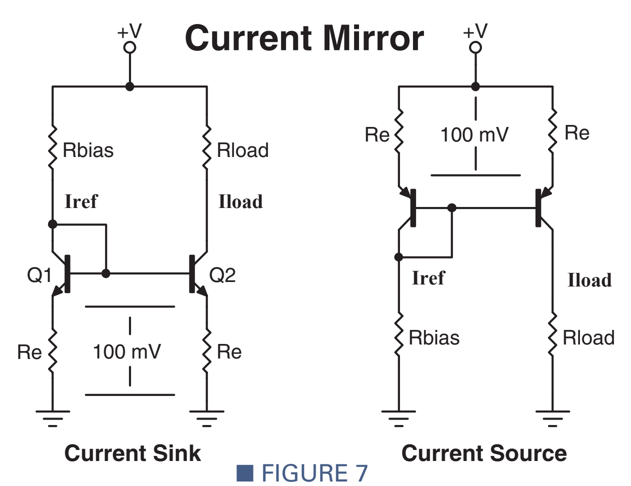
MAILBAG

Dear TJ,

I appreciated your explanation of current mirrors in the July '06 issue, but saw that there was a mistake in Figure 4. The right hand side labeled "current source" must be made with PNPs, not NPNs. By the way, when working with discrete devices, one suggestion I've found helpful is to insert resistors in the emitter leg of each BJT to improve the matching between transistors. This improves the accuracy of the current mirroring and stabilizes the ratio against small differences in temperature between the reference device and mirror device(s). The resistors should be just large enough to produce about a 100 mV drop at the operating current — and must all be the same value.

— Jeff Berwick

Response: *I like your suggestion. In fact, that's how the Figure 4 drawing error occurred. I had 16 different versions of current mirrors on one page — way too big to publish. So I did a simple process of eliminating those I didn't want and reconfiguring the remaining. Guess what? Yep, erased the one I wanted and ... well, you can guess the rest. Figure 7 shows the correction — plus your suggestion. Thanks a lot for your feedback! — TJ*



■ FIGURE 7

www.americanmicrosemi.com/products/search/

www.decodesystems.com/old-ics.html

But seriously, they have all gone to PICs ... or their equivalent. For example, the MK151 clock kit sold by Ramsey and others uses a programmed PIC16C54C-04 microcontroller as its sole IC. So let me put up a challenge to our readers. Come up with a working PIC, Atmel, or Motorola microcontroller and I will see that it is made available to our readers — with compensation for your time, of course. For further details contact me at TJBYERS@aol.com. Thanks!

DESIGN FOR OBSOLESCENCE

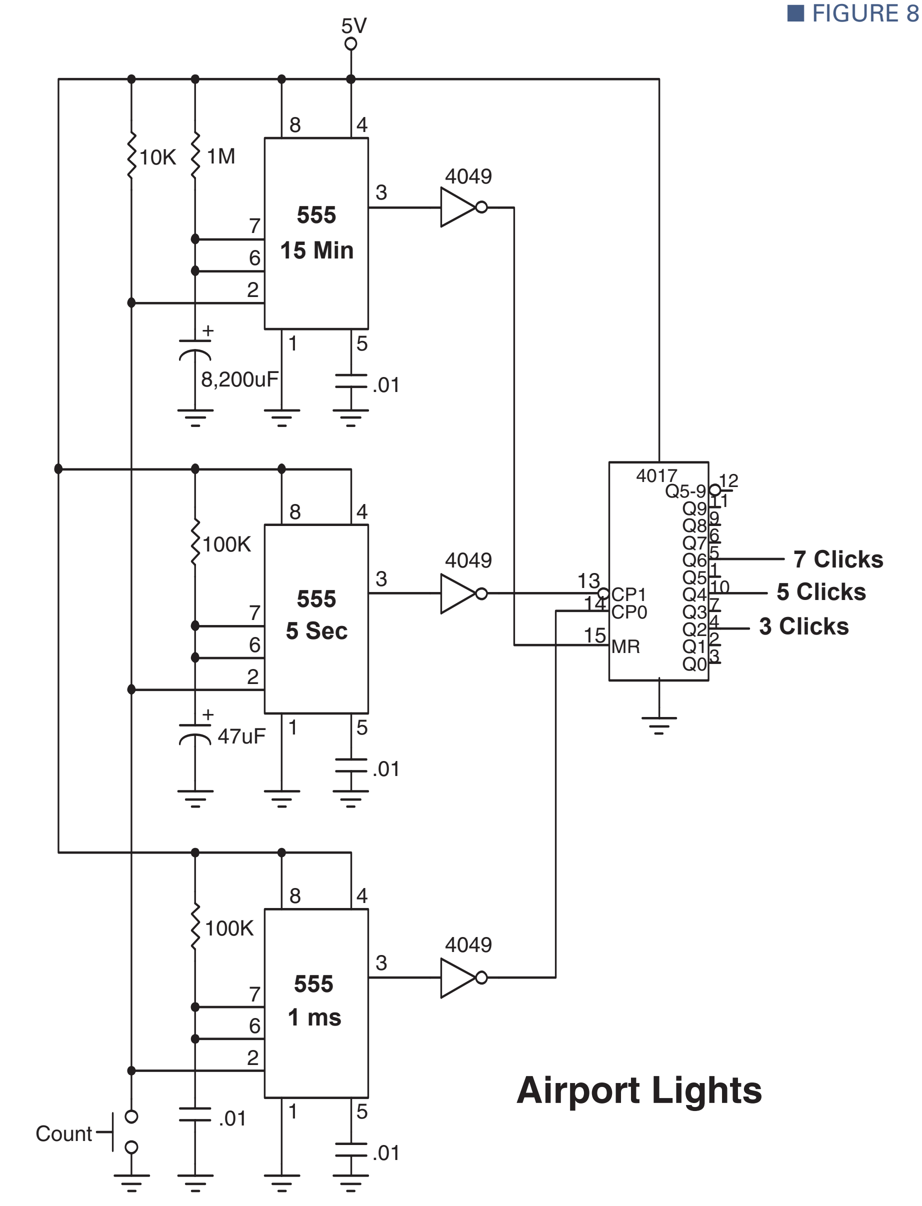
Q I am looking for an Intersil ICM7260 Programmable 59 Timer/Counter. This is an older chip and I have tried several suppliers without luck. So I thought you may have a source.

— Jerry

A Intersil has been acquired by Maxim, and short of a bulk purchase from an obsolete semiconductor jobber, the only place to buy this chip is from **www.maxim-ic.com/parts.cfm/p/ICM7260**. It appears this chip will be in production for the short term. But more and more, microcontrollers are taking over these chores. So be leery when using older ICs in a new design without first checking with the manufacturer to see how long they expect to make and support the product.

For example, as of this writing, Microsoft has pulled the plug on Windows 98 — a stable operating system used by many hobbyists. That means no more updates, no more bug fixes, and no more answers when things go wrong. The same applies to ICs that have run their course.

■ FIGURE 8



CLICK IS ONE, CLICK-CLICK IS TWO ...

Q At night, pilots can control runway lights at small airports by clicking their radio microphone. Seven clicks within five seconds turns on the bright lights. Five clicks is medium and three clicks is low. After 15 minutes, the lights turn off — an obvious way to save electricity. Well, I'm looking for a circuit that will let me control operations via the number of received clicks within five seconds. Ideally, the circuit would have a timer that once activated would shut off after a

COOL WEBSITES

Petals Around The Rose: They say the smarter you, the longer it takes to solve the puzzle. Rumor has it Bill Gates took one full week to solve it. Stupid me solved it in four moves.

http://vincentwoo.com/files/petals

http://crux.baker.edu/cdavis09/roses.html

Urban legends: fact or fiction?

www.snopes.com

certain amount of time.

— Christopher Rust
Maple Grove, MN

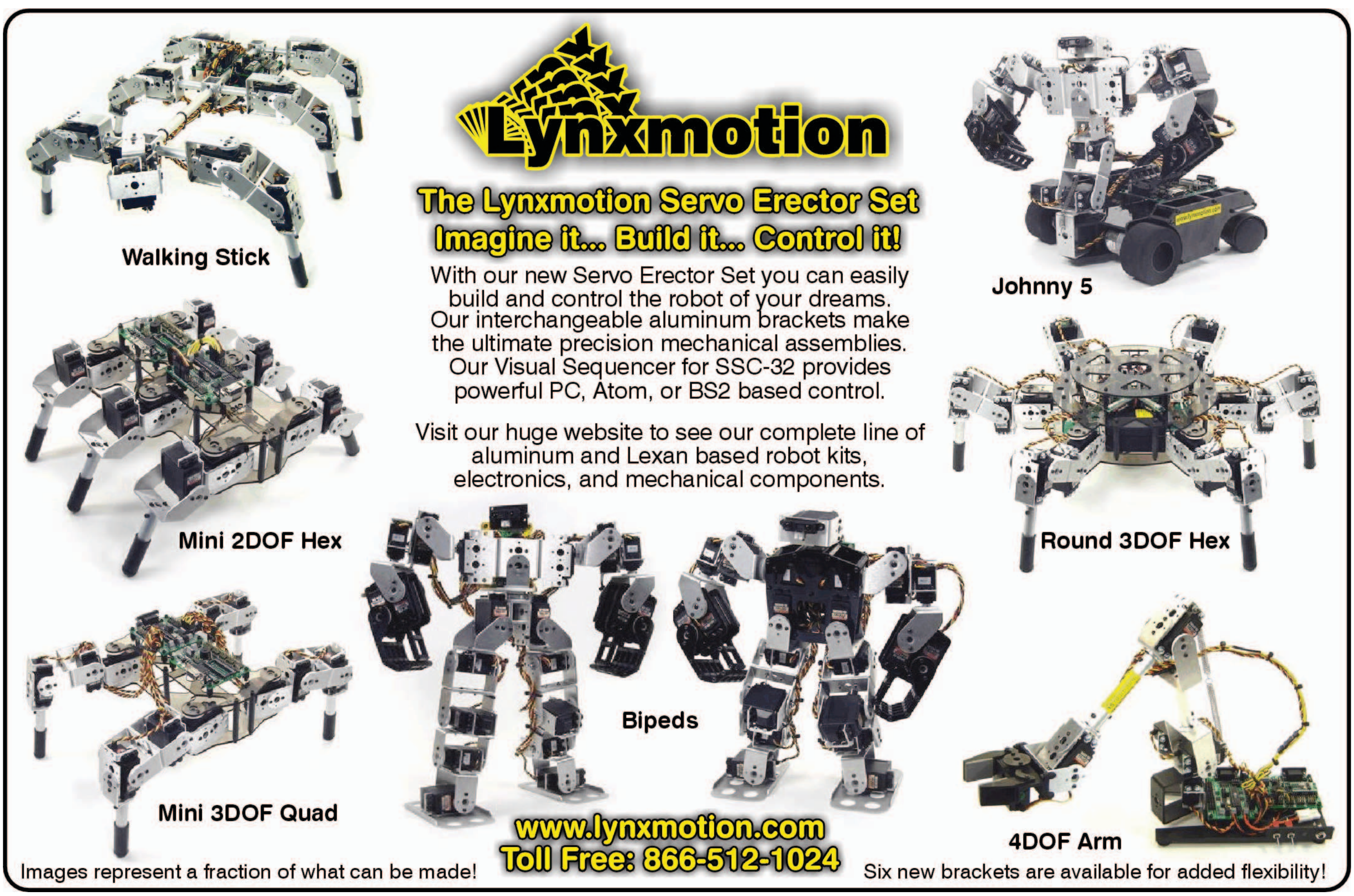
A This brings me back to my childhood, where we used voice commands to operate a model railroad. STOP! (One syllable.) GO FORWARD! (Two syllables.) NOW BACK UP! (Three syllables.) Back then, we used a microphone, tube amp, and a ratcheting relay to make the train obey the "voice" commands. Thankfully, today we have the 4017 decade counter — which is a lot quieter and uses less power. All you need to do is turn on the 4017 for five seconds and count the number of pulses that arrive in that time.

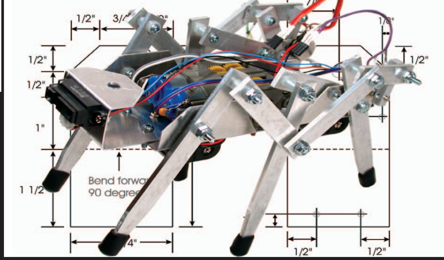
The 4017 will need some help, though, in the way of sequencing timers. See Figure 8 for this discussion. We first assume that the 4017 is on standby, in that the Reset (pin 15) and Clock Enable (pin 13) inputs are both high. For this circuit, I'm using a push-button switch — but any negative-going pulse will do. Push the button once, and the 15-minute and five-second timer go high and turn on the 4017 through the 4049 inverter. This sets the stage for counting — but the clock pulse (pin 14) can't arrive until the 4017 has stabilized. Hence, the 1 ms timer. All clock pulses are delayed by one millisecond.

Once the Reset and Clock Enable inputs are established, let the counting begin. After five seconds, the Clock Enable line goes high and freezes the count. Fifteen minutes later, the Reset input goes high and clears the output, readying the circuit for the next pulse train. As for the outputs — you're on your own. Note that the brightness of the lamps will ripple from dim to bright as the count progresses. If that's not acceptable, use the five-second output for an inhibit signal. Of course, the 15-minute and five-second timers can be adjusted to any periods you wish — and can be reduced to a single chip using a 556 IC. But leave the 1 ms timer as it is.


EMBEDDED LINUX STARTER KIT
FEATURES INCLUDE:
• Linux 2.4 Kernel
• 486-133 MHz based SBC
• 10/100 Base-T Ethernet
• 16MB DOC Flash Disk
• 32MB RAM
• Power Supply
• Carrying Case
• Starting at $399.00
• X-Windows (option)
• RealTime Linux (option)
Imagine running Embedded Linux on a Single Board Computer(SBC) that is 4.0" x 5.7" and boots Linux from a Flash-Disk. No hard drives, no fans, nothing to break. Now your hardware can be as reliable as Linux! If your application requires video output, the X-Windows upgrade option provides video output for a standard VGA monitor or LCD. Everything is included; Ready to Run Linux!
Since 1985 OVER 20 YEARS OF SINGLE BOARD SOLUTIONS
EMAC, inc.
EQUIPMENT MONITOR AND CONTROL
Phone: (618) 529-4525 • Fax: (618) 457-0110 • www.emacinc.com


Going Wireless is Easy!
Mesh Networking
ZigBee™
Bluetooth
with Dip-Switch options
Multi-Channel
900 MHz Wireless RF Modules
LEMOS INTERNATIONAL
Call Toll Free 1-866-345-3667 or email at sales@lemosint.com
www.lemosint.com


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
With our new Servo Erector Set you can easily build and control the robot of your dreams. Our interchangeable aluminum brackets make the ultimate precision mechanical assemblies. Our Visual Sequencer for SSC-32 provides powerful PC, Atom, or BS2 based control.
Visit our huge website to see our complete line of aluminum and Lexan based robot kits, electronics, and mechanical components.
Walking Stick
Johnny 5
Mini 2DOF Hex
Round 3DOF Hex
Bipeds
Mini 3DOF Quad
4DOF Arm
www.lynxmotion.com
Toll Free: 866-512-1024
Images represent a fraction of what can be made!
Six new brackets are available for added flexibility!



■ BY PHIL DAVIS and KEN TAIT

THE SAGA OF THE SILVER BOMBER — PART 1

■ PHOTO 1. Four Silver Bomber Scooters.

OUR ROBOTICS CLUB CONSISTS OF an eclectic and diverse group of people. Like most clubs, some are better at "procurement" than others. Such is the case with our club president, Jerry.

One day Jerry came across some broken Silver Bomber scooters at an industrial surplus house he frequents in town. Talking to the owner, he discovered they had multiple palettes of the stuff lying around. Originally, the owner wanted $35 a scooter — not too bad considering they sold new for $199 plus.

Not being satisfied with this, Jerry kept bargaining and finally promised the owner that the club would buy enough to make it worth his while at $10 each. He agreed and one bright Saturday morning, we adjourned the meeting and headed over to have a look.

The place was a dump ... literally, but in the end we went through many palettes to pick out our treasure. All in all, club members bought 200 scooters that day. One of the club members bought 24. I happened to buy four, as did many of our members. Photo 1 shows my four rather grubby scooters in various states of repair, sitting in my basement. In Photo 2, you can see the rear wheel and the belt drive.

■ PHOTO 2. Rear view of Bomber.

SALVAGED PARTS

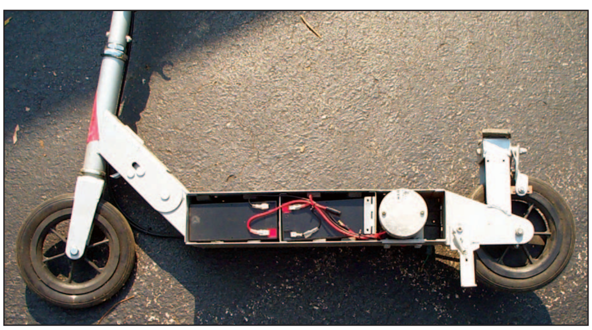
What on Earth, you might ask, do we want with a bunch of Silver Bomber scooters? For the salvage of all the parts, of course. Each scooter had a PM DC motor, timing belt driven drive wheel, timing pulley on the motor, timing belt, two 12V gel cell batteries, and a stack of other bits and pieces. Photo 3 shows a scooter on its side with the battery cover removed and, in Photo 4, you can see most of the useable salvaged parts.

We were dubious about the batteries, but it turned out almost all of them were serviceable and recharged fine, not bad — eight functional batteries, motors, and all the other stuff for $40. The speed control (well it wasn't, really) was just a pair of large switches, one for forward, one for reverse. Pedal to the metal with no real control.

The motors were some of the most interesting parts ... can't have a decent robot if you can't make it drive around. As can be seen from Photo 5, they are Chinese made. It turned out one of our club members, Brandon, had access to a "dyno" and kindly offered to test the motors with the following promising results:

- At 24V, it's about a 200W motor (or ~1/4HP)
- At 36V, it's about a 470W motor (or ~2/3HP)

All in all, I think we did very well and ended up with some good stuff. And so, it all sat waiting to be turned into something we could call a robot ...



■ PHOTO 3. Bomber without battery cover.

HOW TO DRIVE THE MOTOR

Late one night, Ken, Jerry, Tom, and I were at Jerry's shop laser cutting tiles for our upcoming line

following competition (that's another story), when Tom commented that there were no good, inexpensive (I think he used the word cheap) speed controllers to use with these motors. So, as a group, we pointed at Ken and insisted that he design one. Since he also bought some silver bombers (five, I think), it was hard for him to refuse the challenge.

I'll let Ken describe the design process ...

KEN STARTS THE DESIGN PROCESS

I (Ken) bought my Silver Bombers with the intention of either using just the motors or turning them into some kind of tracked robot. A PWM driven controller is a must to build something like this.

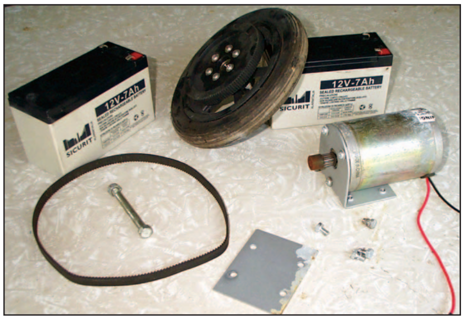
There were no markings on the unit, so the only way to determine what kind of current it needed was to measure it. Using a current limited power supply, I ran it at six volts and determined that the current draw was about 2.5 amps with no load. I measured the winding resistance by carefully rotating the commutator until the lowest possible reading was obtained. This value would give a good indication of what the stall current would be. I measured about .715 ohms.

E=I*R give us 24V/.715 ohms or 33.5 amps. I'll take that as 35 amps for the design. With this in mind, we needed a controller that could safely withstand a 35 amp stall current. I asked some other club members to test their motors and they got similar results.

In the interest of full disclosure, I had already started a project like this some time ago, but was going for something much more sophisticated. I wanted a microcontroller driven unit with full quadrature or servo input that was capable of running some fairly sophisticated PID loops. Knowing this group as I do, I decided we needed something a little simpler to start with — a study H-bridge with simple control and as much protection built in as we could afford.

To give you a little background on me, I am a hardware/software designer with more than 35 years experience in many design venues. Robots have been in my blood since I built my first one at age 12. Designing electronics and software is second nature to me, but some of the intricacies of machining and CNC were a mystery. That's why I joined the group — to trade talents and help others out if I could. Phil and I regularly collaborate on various robotics projects.

■ PHOTO 4. Parts salvaged from a Bomber.



A good design always starts with research. There is no point re-inventing the wheel, and lessons learned by the efforts of others are invaluable in producing a good, repeatable design.

I searched the web and looked at many robotic H-bridge projects. Most had some aspects of what I was looking for, but none quite put it all together in the professional (and sometimes quirky) manner I'm accustomed to.

For the purposes of this article, I'll dispense with all the design equations needed to select parts and properly control the MOSFETs. These are readily available in many places; a quick search will ferret them out if you are really interested in this aspect of the design.

I settled on using the Intersil HIP4081A driver — a common and hardy solution with more than enough drive for a 50A controller. I had talked to the Intersil rep and he assured me it was a solid part and that they intended to support it for some time to come.

Coincidentally, I discovered that this is the same part used in the OSMC controller project and many of the humanoid robot designs in Japan. Couple this to some low Rds on FETs and you have the makings of a reliable driver.

The club members all have different controllers they are experimenting with. Some are using Bascom, others using embedded 'C' code. To accommodate this variety, I chose to provide the following inputs that can be driven by any microprocessor:

1. A master disable to disconnect the FETs.

2. A direction pin.

3. A PWM or brake input. When held low, motor braking action is initiated.

4. A fan enable pin. The fan can be turned off to conserve power when idle. The fan can also be PWM driven to provide fan speed control.

5. An auxiliary relay drive. Can be used to totally disconnect the motors or control brakes if using wheelchair motors.

6. A scaled voltage output for measuring the input voltage.

7. A heatsink temperature output scaled 10 mv/deg C.

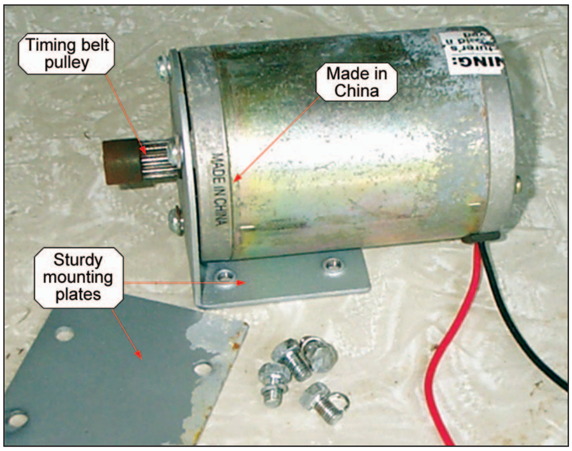
All of the inputs and outputs are isolated (most optically), except the scaled input voltage. Fully isolating this would have made the project complete, but added too much cost.

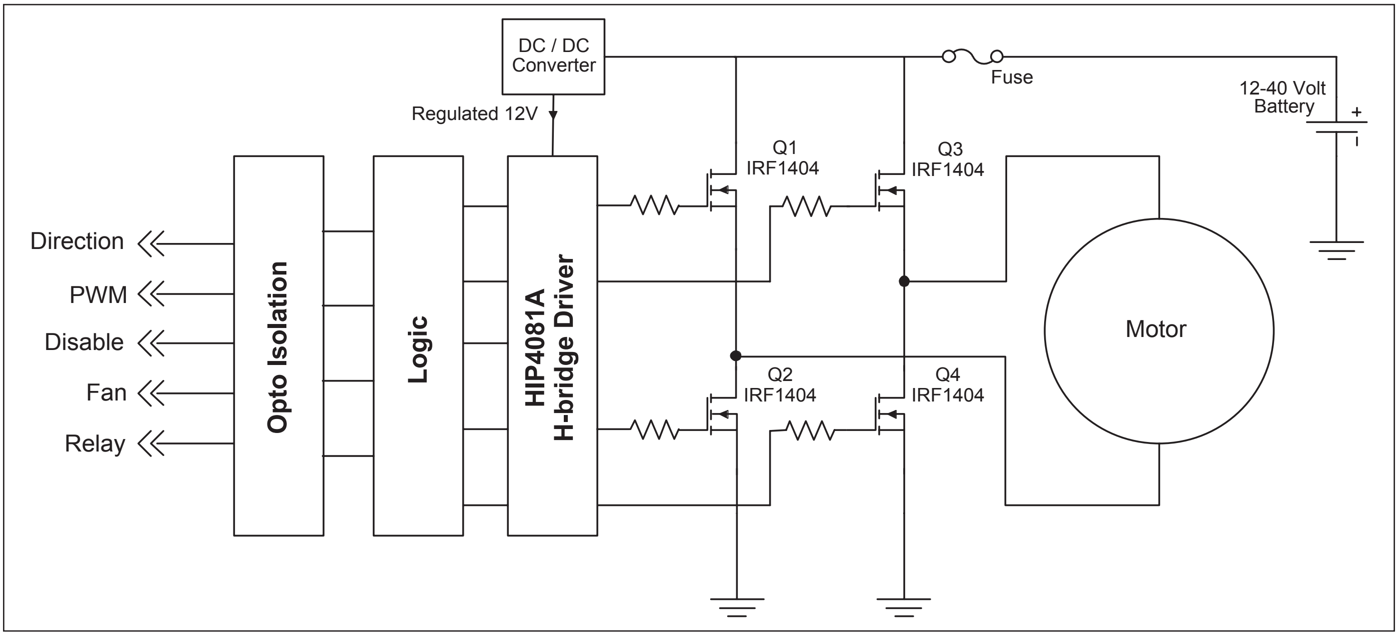
In addition, since it is opto-isolated, the driver can be powered by controllers running on 3.3V power supplies without any level translators.

THE CONTROLLER

The block diagram in Figure 1

■ PHOTO 5. Bomber motor.





■ FIGURE 1. The motor controller block diagram.

outlines the major board sections. From the diagram, four or eight N channel FETs (only four are shown) can be populated depending upon the total drive requirements.

The FETs we are using are rated for 162A each, however this is not really the truth in a real world application. Practically speaking, the maximum power that can be switched is dictated by what the package (in this case, a TO220AB) can safely dissipate. The PWM frequency, duty cycle, motor resistance, and input power level all have a bearing on how the part will perform. In addition, the leads on the package and the lead frame (the internal part that holds the chip) will not pass this current without glowing cherry red ... there is a big difference between pulse power and continuous power operation, something most data sheets forget to mention.

For larger currents, these problems can be mitigated by paralleling FETs together or using a package that will dissipate more heat. The OSMC design uses up to four FETs per leg, however we are only interested in 50A max with adequate cooling, so this will be more than enough.

The driver, HIP4081A contains circuitry to generate the necessary gate drive voltage and logic to ensure that two FETs are never on at the same time causing a condition known as "shoot through" ... smoke to us ... the time delay associated with this switching action is controlled by two external resistors — one per drive section.

The logic section ensures the direction input toggles the correct phases and routes the PWM signal accordingly. The logic is also set up to ensure the driver comes up in a "safe" mode with the motor off if your controller is not powered up.

The isolation section consists of a quad opto-isolator and a single higher speed opto-isolator. The digital signals are essentially at DC levels, so a slow opto-coupler is all that is required. In addition, this part was chosen to present a light load to the controller. Most LEDs require at least 10 mA of current, but this part can be driven by 5 mA or less.

The high-speed coupler is required to correctly pass the PWM signal with a minimum of edge distortion. I left some flexibility here to experiment with higher PWM speeds and different opto-isolators. Running the controller at different PWM

frequencies tends to cause problems with opto-isolators. They have an inherent turn on and turn off delay, which results in some edge shaping. Too slow a response would result in poor control, additional heating, and possible damage. With the part chosen, we should be able to experiment up to at least a 50 kHz PWM rate.

Of course, there is no guarantee that opto-isolating the inputs will prevent your controller from being damaged if you smoke the driver, but I'm hoping you will be hard pressed to do it. I'll leave that to Tim, our resident smoke artist to test ...

The power section uses a DC/DC converter to drop the input voltage down to a steady 12V for the driver, fan, and auxiliary relay without wasting power. An additional three terminal linear 5V regulator provides the logic supply.

Two automotive fuse holders were added to the circuit. If sized properly, these fuses will offer some level of protection from total catastrophe, but aren't a perfect solution. Some might argue that this is a waste of time. We have used this method in many high power switching circuits before and it does offer some protection; at least it will stop a dead short across your battery stack. Nothing generates smoke faster than a locked rotor.

There is NO reverse polarity protection on the battery input. This could have been done with a large diode across the fuses and input, but I didn't want to give up the PC board area for it. I may revisit this idea on the next revision.

At first, I was tempted to try and wire this up on a perf board, but a friend that designs high power circuits advised me not to waste my time. Motor drivers are funny circuits with many circulating ground currents. To do it right, it should be on a PC board, with 4 oz copper layers (see Figure 2 — available on the *Nuts & Volts* website at **www.nutsvolts.com**).

Three days of layout produced a compact package with heatsink and fan, all in a 4.25" x 4.5" x 3" high package. It is a combination surface mount and through hole design. This is due to a combination of factors, including cost and the high currents involved.

The boards are out for fabrication and will be back soon.

(Phil again ...) It's really great to have an electronics wiz like Ken as part of your robotics club; it certainly makes some things a lot easier. Next month, we'll assemble one of the motor controllers and see how it performs by writing some control code and driving one of the motors. Stay tuned. **NV**

RESOURCES

■ The HIP4081A Driver
www.intersil.com/cda/deviceinfo/0,0,HIP4081A.html

■ The OSMC official site on Yahoo!
http://groups.yahoo.com/group/osmc/

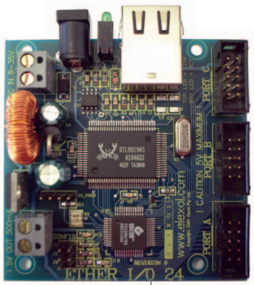
■ International Rectifier, the MOSFETs
www.irf.com/indexsw.html

■ Bascom, for those interested
www.mcselec.com/

NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

ELEXOL ETHER I/O 24



The Ether I/O 24 from Ortech Education Systems is a UDP/IP controlled digital Input/Output module. The module features three eight-bit ports with 5V level signal lines. Each of the 24 lines can be independently programmed as either an input or output.

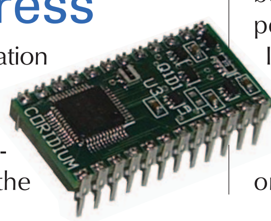
The module connects to any Ethernet network supporting the TCP/IP protocol suite and can communicate with any point on that network. By connecting with an Internet Router, the device can communicate with any Internet connected device. Features include:

- Supports ARP, BOOTP, DHCP, ICMP, and UDP/IP protocols.
- Industry standard 10BaseT Ethernet interface with an industry standard RJ-45 connector.
- 24 independently programmable signal lines with configurable CMOS, TTL, or Schmidt Trigger thresholds and programmable pull-ups per line.
- Easy connection by three 10-way box headers to suit low cost, standard IDC or crimp connectors.
- Integrated switch mode voltage regulator allows power from any 8-32V DC power source.
- Uses 5V 500 mA output to power external interface boards or sensors.
- Compact module measures only 72 mm x 72 mm x 24 mm.
- Advanced configuration allows the modules to automatically scan the input ports and transmit changes directly to another Ether I/O 24 module without host connection or to any Internet port by router connection.
- On-board EEPROM allows all ports to power up in a user programmable state.
- Programmable fixed IP or dynamic IP assignment from a DHCP server.
- Small packet size and connectionless protocol allows for real time sensing and control.
- Can be connected to a wireless network gateway or access point for wireless operation.
- Low power consumption; only 1.1W fully operational.

Applications include home or industrial automation, digital input and output from any networked PC, remote data acquisition and/or alarm monitoring by network or Internet, PC-controlled machines and distributed machine I/O, and remote lighting /power control and/or monitoring.

For more information, contact:
Ortech Education Systems
1119-9th Ave. South
Moorhead, MN 56560
Tel: **218-287-1379**
Email: **Ortner@orteches.com**
Web: **www.orteches.com**

ARMexpress



Coridium Corporation introduces the ARMexpress — the fastest Stamp compatible module. As the name implies, a Philips LPC2106 ARM CPU running at 60 MHz has been mated with a built-in Basic compiler to achieve high performance in a small package. The 600 mil, 24 pin pluggable device is priced at $69.

Because the Basic source is compiled, the performance exceeds 10 million Basic lines per second, and update rates for control of the 16 TTL compatible I/Os exceeds 1 MHz. This performance is comparable to programs written in C. 40K is available, enough for 3,000 lines of user program, and 40K for data. 4K of Flash is also available for writing. In addition to the native 32 bit integers, string processing has been added. This includes concatenation, STRCMP, LEN, LEFT, and RIGHT functions. The real time clock is also integrated into the Basic with pre-defined variables SECOND through YEAR. In addition, a free running microsecond TIMER is available to the user.

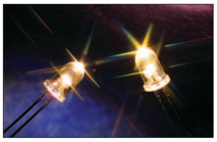
The use of Basic makes the module quick to master and easy to use. It is perfect for solving the problems of building a quick prototype, coming up with the glue for factory automation problems, as a flexible lab bench tool, or fun for hobbyists. The user interface will integrate with your favorite text editor, and the simple but intuitive control program is easy to learn. The compiler is incremental and gives the feel of a scripting tool. It also supports displaying variables immediately and running single statements immediately without recompiling the Basic program.

To ease control of external devices, support for the popular SPI, I^{2}C, 1-Wire, and ASYNC protocols are built in. This allows control of serial peripherals at rates up to 800 Kbits/sec. In addition, library support includes PWM, SHIFTIN, SHIFTOUT, PULSIN, PULSOUT, RCTIME, and FREQOUT. These functions all have a one microsecond resolution.

Programming the device uses two RS-232 compatible pins that can also be used for full duplex operations at 19.2 Kb. The module accepts an unregulated voltage supply of 5V to 12V, as the ARMexpress has onboard 1.8V and 3.3V regulators. The $149 evaluation kit adds USB connectivity. When connected to a USB port, power is supplied by the host PC. The evaluation kit will also operate with the supplied external power source. There is a generous five square inches of prototyping area which will accommodate 0.1″ headers, DB-style, or 3.5 mm terminal blocks. The board layout is also compatible with the Hammond 1455J enclosures.

For more information, contact:
Coridium Corporation
Tel: **800-478-9020**
Web: **www.coridiumcorp.com**

L200-XIW SERIED LED



LEDtronics® announces the super bright L200-XIW Series LED. XIW series Super Bright LEDs are significantly brighter than standard LEDs. These intense LEDs come in the standard 5 mm size allowing them to be useful and implemented in many common applications. They are ideal in applications such as mini lamps, standard electrical based bulbs, displays, theater décor bulbs, telecommunications data products, push button switches, safety area lighting, street lighting, medical equipment lighting, or any application where you need the warm glow of 3,000K color.

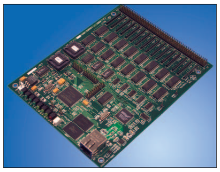
The XIW Super Bright LED comes in a water clear dome lens housing. This clear housing allows for a 20° viewing angle that emits a beautiful and bright 3,000K warm white color. This tiny LED puts out three times the normal light at 14,000 MCD while using only 3.2 to 4 volts DC. Other viewing angles are available for OEM customers.

Solid-state design renders LEDs impervious to shock, vibration, frequent switching, and environmental extremes. With an average life span of 100,000-plus hours (11 years), LED lamps operate more than 20 times longer than the equivalent incandescent lamp. LED lamps produce almost no heat and require 80%-90% less operating power than equivalent incandescents, making them as friendly to the environment as they are to the operating budget.

Pricing for the super bright L200-XIW Series LEDs is $18 for a package of 10 LEDs. Large quantity discounts are available. Availability is stock to four to six weeks for special requirements. L200-XIW Series LEDs carry a one-year limited warranty.

For more information, contact:
LEDtronics, Inc.
23105 Kashiwa Ct.
Torrance, CA 90505
Tel: **800-579-4875** or **310-534-1505**
Fax: **310-534-1424**
Email: **webmaster@ledtronics.com**
Web: **www.ledtronics.com**

ETHERNET TO DIGITAL INTERFACE BOARD



ICS Electronics has announced a new Ethernet to Digital Interface Board for controlling digital devices over a company network or over the Internet. Called the Model 8013, this new interface board provides 128 digital I/O lines that the user can control from any computer with an NIC interface or from a TCP/IP network. Typical applications for the 8013 are interfacing digital devices, controlling relay matrices, or acting as the Ethernet interface for a test chassis or instruments.

The Model 8013 is an Ethernet to Parallel Interface that provides 128 parallel I/O lines that can be configured as inputs or outputs in eight-bit bytes. Data transfer can be done by a combination of three methods depending upon the needs of the devices connected to the 8013. First is by transferring data directly to or from a specific byte, second by strings of data characters to or from multiple bytes, or third, by setting or reading individual bits in a byte. Handshake lines are provided for synchronizing the data transfers or for latching data into external devices.

The 8013 can also monitor 15 input lines and generate VXI-11 Service Request messages when an enabled line changes state. The 8013 is an IEEE-488.2 compatible interface that responds to the 488.2 common commands and uses SCPI commands to configure its digital interface. Users can customize the 8013's IDN message to integrate the 8013 into their systems. All settings are saved in Flash memory.

The Model 8013 is a VXI-11.3 compliant interface. VXI-11 is a communication standard developed by the VISA consortium in 1995 in conjunction with the VISA Specification. VXI-11.3 is a sub-standard that covers TCP/IP-to-Instrument servers like the 8013. Communication with the 8013 is via VXI-11 RPC protocol over a TCP/IP network.

The 8013 can be controlled several ways: The Model 8013's VXI-11 Service can be accessed by LabVIEW, VEE, Visual Basic, and C language application programs that make VISA calls by selecting the 8013 as the TCP/IP resource. Both Agilent and National Instruments provide VXI-11.3 compliant VISA libraries. Linux, Unix, and other programmers who do not want to use a VISA library can access the 8013's VXI-11 Service by RPC calls from the application program. The VXI-11 Standard includes the necessary RPCGen header files for adding RPC calls to any program. ICS provides a free VXI-11 keyboard program which lets users with a WIN32 computer interactively control the 8013 and other VXI-11 instruments without having to write a program.

ICS's 8013 Ethernet-to-GPIB controller has several unique features: First, the 8013 is 100% VXI-11.3 compliant which is an open communication standard. The 8013 supports reverse channel Service Request messages to alert the client application when an event occurs. The 8013 also supports multiple clients as part of its standard firmware. The 8013 is a RoHS compliant assembly for use in products aimed at the European

market. The 8013 is physically interchangeable with ICS's 4813 GPIB and 2313 serial-to-parallel interface cards. All three interface cards support the same command set, so switching interfaces has minimal program impact.

The Model 8013 ships with ICS's VXI-11 Keyboard utility program and Configuration utility. The Configuration utility lets the user set the 8013's IP Address mode, its IP Address, COMM Timeout, KeepAlive, and Interface Name. The VXI-11 keyboard lets a user interactively control the 8013. Both utility programs run on a WIN32 PC.

Pricing for the Model 8013 is $495 each is quantities of one to four units, FOB Pleasanton, California. Delivery is four to five weeks ARO.

For more information, contact:
ICS Electronics
7034 Commerce Cir.
Pleasanton, CA 94588
Tel: **925-416-1000**
Web: **www.icselect.com**

DS1921 TEMPERATURE LOGGER KIT

The DS1921 included in this new kit from Logic Controlled Technologies is a complete self-contained temperature data logger measuring only 17.35 mm wide and 5.89 mm thick. It has an accuracy of ±1°C from -30°C to +70°C. A built-in, real-time clock and timer has an accuracy of ±2 minutes per month. The logger automatically wakes up and measures temperature at user-programmable intervals from 1 to 255 minutes. It logs up to 2,048 consecutive temperature measurements which are recorded in protected nonvolatile random access memory.

A long-term temperature histogram is also recorded with a 2.0°C resolution. There are programmable temperature-high and temperature-low alarm trip points available. The device records 24 time stamps and durations when temperature leaves the range specified by the trip points. 512 bytes of general-purpose read/write nonvolatile random access memory is available for user programming of various information such as placement location information.

The DS1402D-DR8 Blue Dot Receptor has solder points on the solder side so you can hardwire other one-wire networked devices. You can also use a phone jack duplicator and add more sensors via RJ-11 phone cabling. The FOB, K-CLIP, and PolyTag provide a temporary reusable fastener and labeling system for the DS1921G sensor.

For more information, contact:
Logic Controlled Technologies
PO Box 722100
Houston, TX 77272-2100
Tel: **832-877-2189**
Web: **www.logictechs.com**

WHAT'S NEW IN THE FORUMS?

SERVO servomagazine.com www.nutsvolts.com

SERVO MAGAZINE — NUTS AND VOLTS

T&L Publications, Inc. BULLETIN BOARD

If you haven't been to the *Nuts & Volts* forums lately, you've likely missed out on a lot of great discussions. There are over 3,500 registered users and over 37,000 posts covering every electronics topic imaginable.

Here are some recent topics that have been in discussion:

- Plasma TVs
- Electric Cars
- Building Your Own CPU From TTL Chips
- Invisible Infrared Lamps
- Basic Power Supply Concepts
- Hacking A SUN SWITCH Street Light Photocell
- Maximizing Radio Efficiency
- Programming A Microchip To Speak Text
- Goodbye RFID

Current forums include:

General Electronics Discussion

Computers

Robotics

Programming

New ➥ **Up For Grabs**

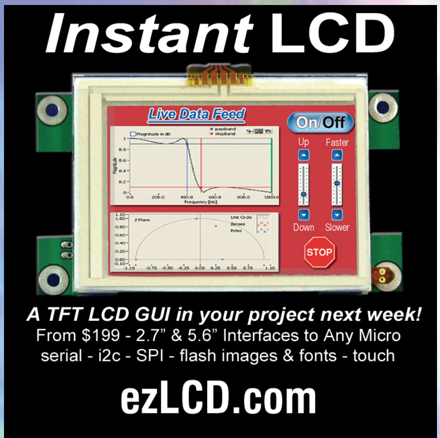
Try out our new **Up For Grabs** forum. It's the place to **post any electronics items for sale, trade, or to give away to a good home**. It's for *private party (personal) items only*. No commercial vendors allowed. Oh yeah, and ***it's FREE***.

So, don't throw out that junk you're not using anymore — it may be just what a budding electronics hobbyist needs to get that project finished!

Check it out for yourself today!

www.nutsvolts.com

SHOWCASE



Instant LCD
Live Data Feed
On Off
Up Faster
Down Slower
STOP
A TFT LCD GUI in your project next week!
From $199 - 2.7" & 5.6" Interfaces to Any Micro
serial - i2c - SPI - flash images & fonts - touch
ezLCD.com

CASH BUYERS
OF ELECTRONIC EXCESS
GREENCHIP
CALL 1-888-GRN-CHIP
476-2447
or visit:
www.greenchiponline.com
Celebrating 10 Years in Business

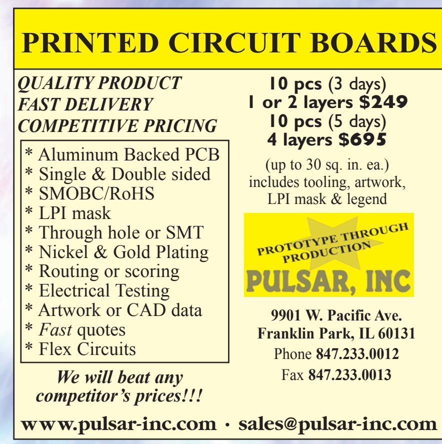
HG139 HDTV Pattern Generator
Great tool for HDTV & Home Theater installer and enthusiast
NEW
Perfect for repair, calibration, and testing of HDTV and other video equipments
MSRP $399
www.GMETechnology.com
10 MHz DDS Signal Generator
NTSC & Computer Monitor Tester
www.gmetechnology.com
TEL: (909)865-1998 FAX: (909)865-2793
Visit our website for other test equipment products
GME TECHNOLOGY

5mm White water clear LED 3.5V 10,000 mcd (AB287) $0.56
20 second voice recorder/playback module. (pre-assembled) (A96010) $6.60
The Velleman Personal Scope is a complete 10MS/s portable oscilloscope. (HPS10) $129.
Thousands more items on our web site!
Electronics123.com
Tel: 1-888-549-3749
102 E. Park Ave, Columbiana OH 44408

LITEXPO™
Digital Video Player Board
MMV-80-USB
$79/ea
(monitor not included)
BUILD YOUR OWN E-FRAME
- Ideal for portable slide show & presentation
- Plays MPEG 1, 2, 4 video, JPEG image & MP3 on TV or any regular video monitor
- USB communication interface
- CF card, A/V output port & remote control feature optional
Tel: (847) 202-0010
WWW.LITEXPO.COM

IV PRD • KIT BARE
$19.00 m.o., kit POSTP'D USA. AVAIL • JUNE 06 Your 5vdc & the IV BB5 w/14.74mHz XTAL, HEX DISPLAY, 8 KEYS, PRE-PROGRAMMED w/ OS+ MONITOR+APP. HAS RS 232, SPI, ISP + 35 I/O
IV POB 245, NJ, 07463

www.Primecell.com
Battery rebuilding service
Dead Batteries ? Don't toss them.
Send them to us - our rebuilds are better than original specifications.
Tools
Hilti Skil
Milwaukee
Panasonic
B&D DeWalt
Makita All
2-36 Volts
Electronics
Bar Code
Scanners
Surveying
Printers
Laptops
Photography
Radios
APELCO
UNIDEN
G.E. ICOM
KENWOOD
MOTOROLA
MIDLAND
MAXON
YAESU
ALINCO
Visit www.primecell.com for important details
24 Hr Secure recorder tel-fax (814) 623 7000
Quotes email: info@primecell.com
Cunard Assoc. Inc. 9343 US RT 220 Bedford PA 15522



PRINTED CIRCUIT BOARDS
QUALITY PRODUCT
FAST DELIVERY
COMPETITIVE PRICING
10 pcs (3 days)
1 or 2 layers $249
10 pcs (5 days)
4 layers $695
(up to 30 sq. in. ea.) includes tooling, artwork, LPI mask & legend
* Aluminum Backed PCB
* Single & Double sided
* SMOBC/RoHS
* LPI mask
* Through hole or SMT
* Nickel & Gold Plating
* Routing or scoring
* Electrical Testing
* Artwork or CAD data
* Fast quotes
* Flex Circuits
PROTOTYPE THROUGH PRODUCTION
PULSAR, INC
9901 W. Pacific Ave.
Franklin Park, IL 60131
Phone 847.233.0012
Fax 847.233.0013
We will beat any competitor's prices!!!
www.pulsar-inc.com · sales@pulsar-inc.com

ActiveWire® USB
Simple USB Interface!
• Works with MacOS 8/9, Win98/2K/ME/XP FreeBSD and Linux!
• 24Mhz CPU core with USB
• Firmware downloadable via USB
• 16 bit parallel Input/Output
• See web-site for add-on boards
• All drivers, manuals, demos are on our web-site for immediate download!
$59 plus shipping
ActiveWire, Inc.
www.activewireinc.com
ph +1.650.465.4000
fax +1.209.391.5060

10 Year Olds Can Now Hand Solder ANY Surface Mount Component - Even BGA
AND SO CAN YOU!
www.schmartboard.com

ANDRE LAMOTHE'S
XGAMESTATION
LEARN STEP-BY-STEP HOW TO BUILD AND DESIGN YOUR OWN VIDEO GAME CONSOLE!
Design inspired by the Atari 2600, Apple II & Commodore 64!
Huge eBook Included!
INCLUDES:
• Great for Hobbyists AND Students!
• The Fun Way to Learn Embedded Systems!
• Fully Assembled XGS Micro Edition Unit!
• Complete Software Development Kit!
• Tools, Demos & Utilities!
• eBook on Designing the XGS Console!
• Cables and Power Supply Included!
Powered by the Ubicom SX52 @ 80 MIPS and Parallax SX-Key Compatible!
COMPLETE KIT
WWW.XGAMESTATION.COM
(925) 736-2098
SUPPORT@NURVE.NET

QKITS.COM
PANEL MOUNT OSCILLOSCOPE
$174.95 (US dollars)
Measurements: rms, dB(rel), dBV, dBm
Direct Audio power measurements
6 display modes
Input impedance: 1Mohm/30pF
Connection of internal or external signals
Bandwidth: 2MHz
Sampling: 10MS/s
Power Supply: 9VDC or 6VAC/300mA
1-888-GO 4 KITS
GREAT PRICES, GREAT SHIPPING RATES

THIS MONTH'S PROJECTS

LEVEL RATING SYSTEM

To find out the level of difficulty for each of these projects, turn to our ratings for the answers.

●○○○ . . . . Beginner Level

●●○○ . . . . Intermediate Level

●●●○ . . . . Advanced Level

●●●● . . . . Professional Level

This circuit produces a sine wave, square wave, and triangle wave of nine volts amplitude.

There are three sections to the circuit: power supply, sine wave generator, and the square wave and triangle wave generator.

NINE VOLT FUNCTION GENERATOR

The power supply uses a power transformer to drop 117 VAC down to 12 VAC at 60 Hz. A doorbell transformer could be used for this purpose since the current requirement for this type of circuit is low. In fact, a one amp fuse is used since the current rating of the 555N timer IC and 486 operational amplifier are 1.5 amps. Switch 1 — a single pole single throw switch — is the on/off switch, the state of which is indicated by LED D5. R1 limits the current of the LED to protect it.

Diodes 1 through 4 are generic rectifier diodes in the common full-wave bridge circuit configuration. L1 and C1 form a simple RCL filter to smooth out the rectified DC. The output of the power supply is supplied directly to pin 6 of the LM386 and pin 4 and 8 of the 555 timer ICs.

The square wave and triangle wave IC is a 555N timer, IC2 on the schematic. A triangle waveform is obtained at pin 1 of the 4,700 µF ceramic disc capacitor. Since the circuitry is based on AC, I would prefer using electrolytic capacitors. However, I have not had trouble using CD capacitors. If pins 2 and 6 were not attached to the 4,700 µF capacitor, then this circuit would produce a square wave at pin 4 only.

The 1M potentiometer varies the frequency. C6 allows a path for AC to ground, while blocking any stray DC. The value of C7 varies the frequency, just as the value of the 1M potentiometer does. Pin 6, Vs, is the feedback loop to the inverting input, pin 2. Gain, pin 1, is directly fed to ground. Pin 7, bypass, connects between both

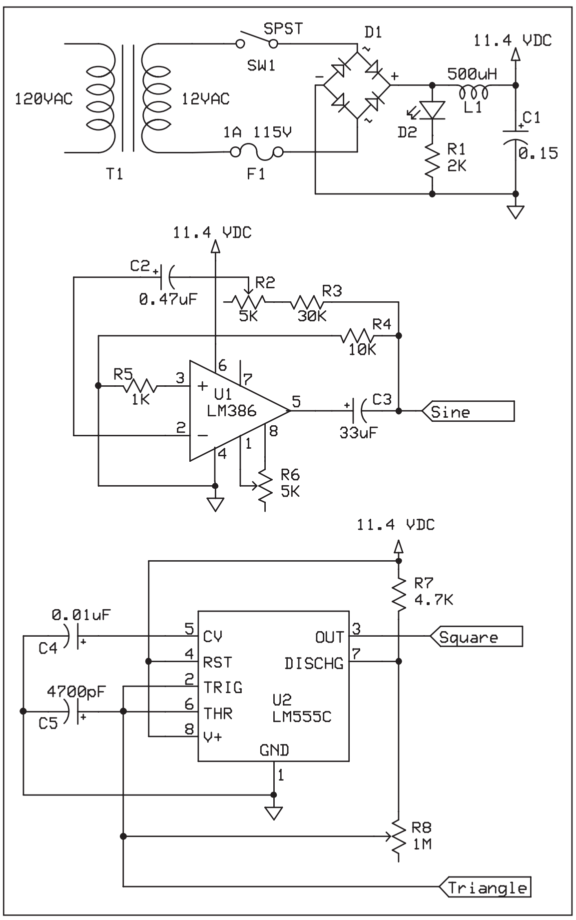


FIGURE 1

the frequency-varying components — C6 and R2. R3 (4.7K), connects the circuitry of this chip to the sine wave circuitry. Ground is common to all parts of the circuit, of course.

For the sine wave portion, an LM386 operational amplifier is used as a non-inverting (negative-input grounded, positive input used), integrating-gain amplifier configured to produce and amplify self-produced sine wave oscillations of a frequency of about 50 kHz to 100 kHz. Pin 5 is the sine wave output and pin 6 is the rectified voltage input. Gain feedback is adjusted via R4, a 5K potentiometer. C8 prevents DC from contaminating the sine wave output. Feedback after this capacitor goes to pin 3 (+ input) through R5 and R6, which provide 11K resistance. The amplitude of the sine wave is varied by R7, a 100K potentiometer, and fed back to pin 2, which is the negative input. DC isolation and chip bias are provided by the 0.47 µF capacitor C9.

Under test, I have noticed that the frequency range varies by about 10% between different circuits that I have built. The sine wave tends to have a small degree of distortion. The triangle wave is sharp and crisp, as well as the square wave. The square wave is susceptible to distortion when capacitively loaded by an oscilloscope. I suggest calibrating this function generator with an oscilloscope and noting the degree of any distortion which, unfortunately, tends to creep around the circuit a bit. **NV**

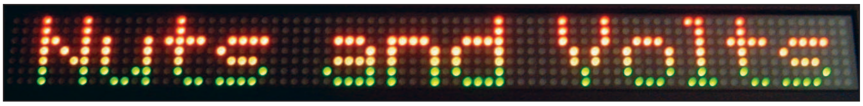
PARTS LIST

All resistors are 1/8 unless otherwise noted

ITEM	DESCRIPTION
❑ R1	2K
❑ R2,R6	5K pot
❑ R3	30K
❑ R4	10K
❑ R5	1K
❑ R7	4.7K
❑ R8	1M pot
❑ C1	.15 µF electrolytic
❑ C2	0.47 µF electrolytic
❑ C3	33 µF electrolytic
❑ C4	0.01 µF
❑ C5	4,700 pF
❑ U1	LM386 op-amp
❑ U2	LM555C timer
❑ D1	Bridge rectifier (RadioShack 276-1181)
❑ D2	LED
❑ T1	120V/12V transformer
❑ L1	500 uH inductor
❑ F1	1A 115VAC fuse
❑ SW1	SPST switch

CONTROL YOUR WORLD

PART 3 — Interface Your PC to an LED Sign



I was doing a robotics show and needed a way to both attract contestants and display contest results. I had seen a couple LED signs and decided one would be perfect for this kind of application. My only requirement was that the cost had to be reasonable and the sign needed to be easy to transport and hook up.

I needed a sign with an RS232 interface so I could display real time contest results.

The most prevalent LED sign was one called the BetaBrite. They were stocked at most office supply stores. The best bargain was when I found them on sale at one of my local warehouse stores. They cost me less than $100 each. You can get more information about these signs at **www.BetaBrite.com**.

The signs are manufactured by a company called Adaptive Micro Systems, Inc. They manufacture several signs — everything from single line indoor signs to multi-line outdoor signs.

The BetaBrite Sign comes with an AC adapter for power, IR remote control, and a CD that contains a simple program for uploading messages to the sign from your PC (see Figure 1).

■ FIGURE 1

Being the kind of person I am, I was compelled to write my own interface. I figured that once I learned the protocol, I could interface to the display with microcontrollers, laptops, or my Pocket PC.

Most of these signs come with an RS232 interface, which is important if you want to interface with devices like Pocket PCs or microcontrollers. If this is your goal, don't purchase a USB controlled sign. The BetaBrite sign comes with a 25 foot, nine-pin RS232 connector and is set up as a DCE device so this connector can be plugged directly into the PC.

If you want to connect it to a Pocket PC, you will need a null modem and gender changer or a Bluetooth RS232 adapter. For connection to a microcontroller, I have created an application note that can be found on the Kronos Robotics website at **www.kronosrobotics.com**.

Research the Protocol

I started researching the interface protocol by searching the Internet. You can find a great deal of information on various websites, but the best source is from the

Adaptive website at **www.adaptivedisplays.com**.

The signs use a protocol called Alpha Sign Communications Protocol. You can find a complete description of the protocol at **www.amsi.com/Pages/97088061.htm**

The protocol can be a bit complicated as it supports networking devices. What I'm going to do in this article is exploit a feature of the protocol to create a very simple interface.

Programming Protocol

I have been doing PC and Pocket PC development for some time now and, a while back, I decided to create a simple development platform that would make interfacing to devices and robotics much quicker and easier. With the help of other engineers, we came up with a development platform called Zeus. Zeus has both desktop and Pocket PC versions available.

I have created a version of Zeus just for *Nuts & Volts* readers. ZeusNV will let you debug, compile, and test the code in this article. You can even create a stand-alone executable that you can distribute to others.

ZeusNV has one requirement. You must have the .Net Framework installed on your Windows machine.

The code we used is quite simple and can be adapted to just about any programming language that supports RS232 communications.

The Alpha Protocol

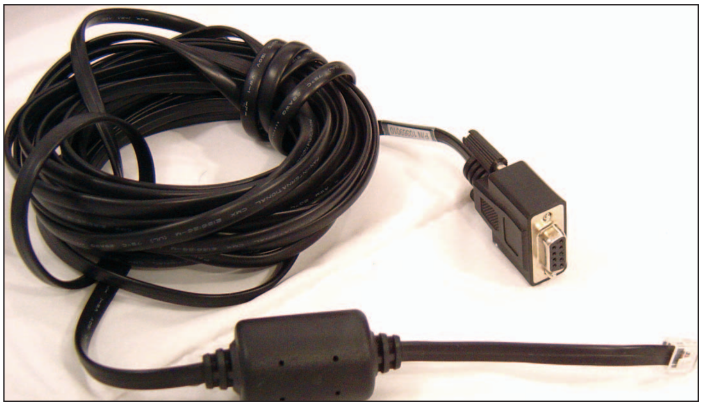
The Alpha protocol has some pretty advanced features such as networking and message labeling. It also supports uploading graphic images call DOTS. In order to make the explanation and examples as simple as possible, we will drastically simplify the interface by using certain simple aspects of the protocol.

Before we send a single byte to the sign, we need to open up a comport with the correct settings. The protocol supports the following baud rates: 1200, 2400, 4800, and 9600 with the following settings: 7 Bits, 1 Start, 2 Stop, and Even Parity. We use the Zeus ComOpen command to open the com port.

```
Const Ch1 1
ComOpen(Ch1,baud=9600,port=1,parity=2,bits=7,stop=2)
```

Zeus can control five Async connections at once. These are called channels; hence, the Ch1 constant. You will need to change the port=1 setting if you are using anything else. For instance, if you are using Bluetooth on an IPAQ Pocket PC to connect to a Bluetooth RS232 connector, you will use port=8.

Once the port is open, we have to tell the sign we are ready to communicate with it. I call this the ready header.



■ FIGURE 2

```
ComOutput Ch1,chr(0)+chr(0)+chr(0)+chr(0)+chr(0)
ComOutput Ch1,chr(1)
ComOutput Ch1,"Z00"
```

We start with five null characters (Value 0). This sets the sign's baud rate to the rate you are transmitting. (Note that if you are dealing with a language that can't send Null characters, you may also send five 1s.) We then send an SOH (Start Of Header, Value = 1) character. After this, we can do many things but, in our case, we are going to send "Z00." This tells the sign that we are addressing all signs on the bus.

The Alpha protocol supports sending checksums. However, to make the interface simpler, we are not going to utilize this feature.

With the header sent, we can now send a command. The Alpha protocol supports several command types, but we will only be using a couple of them. In this case, we are going to send a Priority Text Message. What is neat about this message is that it bypasses the internal file system so it's simple and fast. The down side is that you can only send 125 bytes, but since we are going to be sending real time messages, this is perfect.

To send a Priority Text Message, you start by sending an STX (Start Text, Value = 2) character. We then send a command code of

■ FIGURE 3

PROGRAM 1

```
'Simple BetaBrite Priority Message Sample
func main()

 'Open the ComPort
   Const Ch1 1  'Zues Channel 1
   ComOpen(Ch1,baud=9600,port=1,parity=2,bits=7,stop=2)


 'Tell the Sign we are ready to comunicate
   ComOutput Ch1,chr(0)+chr(0)+chr(0)+chr(0)+chr(0)
   ComOutput Ch1,chr(1)
   ComOutput Ch1,"Z00"

 'Send Priority Text Message
   ComOutput Ch1,chr(2)+"A0"
   ComOutput Ch1,"Nuts and Volts"
   ComOutput Ch1,chr(4)

   CloseCom Ch1

endfunc
```

WEBLINKS

■ KronosRobotics website
www.kronosrobotics.com

■ KRMicros website
www.krmicros.com

■ Adaptive website
www.adaptivedisplays.com

■ BetaBrite website
www.BetaBrite.com

"A" and a File label of "0."

ComOutput Ch1,chr(2)+"A0"

This tells the sign that we are sending a Priority Message.

You then send your message text followed by an EOT (End of Transmission, Value = 4)

ComOutput Ch1,"Nuts and Volts"
ComOutput Ch1,chr(4)

Once the EOT character has been sent, the message will display. By default, the colors and display modes will be in auto format so you will see the message change. We will change this setting later.

■ FIGURE 4

CloseCom Ch1

Now, wasn't that simple? Program 1 shows the complete code example we just discussed.

We are going to add a few new features to the program, so in order to simplify the code, let's take all the current code and place it in a function called BBSendText. We will set the function up so that we can pass the com port and the text of the message to be displayed. Then all we have to do is call this function each time we want to send new data.

We are also going to add a function called BBSetTime. This function lets you send a four-character string to set the current clock located inside the sign. Yep, that's right, the sign has its own clock. There is even a control code that you can insert into your text and the current time will be substituted for that code.

BBSetTime will open the com port, send a Special Function command, and the four-digit time string (24 hour format). The function will then send the EOT code and close the com port.

Note that you only need to set the time once; then you can comment the code out. The clock will keep the current time even when new commands are sent. Once power is removed, the clock will stop. When power is restored, the clock will start again where it left off and the last message you sent will be displayed.

Zeus will automatically shut down the comport for you but it's always smart to do it yourself with the CloseCom command.

You will notice that now all we need to do is use the BBSendText along with various pause commands and we can display whatever we want to in real time.

Notice the second BBSendText function call just after the **Loop** label in Program 2. Change it to:

BBSendText(1,chr(19))

What we are doing is sending a control code that will tell the sign to display the time. In this instance, the control code is 19. Other commands may require more than one sequence of control characters.

Now for Program 3. We will add a very powerful function to the previous program. This function is called ConvertCodes and will take our text message, along with special commands, and create some rather cool effects. You can now control the colors and scrolling effects of the messages.

Here is a list of the commands that you may insert into your message:

Colors
[RED] [GREEN] [AMBER] [DRED] [DGREEN] [BROWN] [ORANGE] [YELLOW] [RAINBOW1] [RAINBOW2] [COLORMIX] [COLORAUTO]

Fonts
[SMALL] [SMALLWIDE] [LARGE] [LARGEWIDE] [FANCY] [LARGE3D] [LARGEWIDE3] [LARGEWIDE4] [PROP] [FIXED]

Roll
[ROLL UP] [ROLL DOWN] [ROLL LEFT] [ROLL RIGHT] [ROLL IN] [ROLL OUT]

Wipe
[WIPE UP] [WIPE DOWN] [WIPE LEFT] [WIPE RIGHT] [WIPE IN] [WIPE OUT]

Other
[ROTATE] [ROTATESMALL] [TIME] [HOLD] [AUTOMODE] [FLASH] [TWINKLE] [SPARKLE] [SNOW] [INTERLOCK] [SWITCH] [SLIDE] [SPRAY] [STARBURST]

Graphics
[SLOTMACHINE] [NOSMOKING] [DRINK] [ANIMAL] [FIREWORKS] [GRAPHIC1] [BOOM]

Speed
[SPEED1] [SPEED2] [SPEED3] [SPEED4] [SPEED5]

To insert the effect, just add it to your messages, as in:

BBSendText(1,"[ROTATE]Nuts and Volts")

Program 3 is a bit too large to include here, but it is included with all the source code provided in this article.

There is a lot you can do with one of these signs. The animated text and graphics will be sure to attract attention wherever you decide to use them, but the real power is going to be real time updates.

As a bonus, I have created a program called BetaBrite TNG shown in Figure 3. It is a cool little program that will make it very easy to set up messages for your sign. There is both a desktop PC and Pocket PC version included. Be sure to check the KRMicros and Kronos Robotics websites for all the source code, as well as some compiled.

Going Further

The BetaBrite sign comes with its own software so why would we want to create our own interface? Well, for a couple of reasons. The most important one is we now have the ability to display real-time information. We could use Zeus to go out and query a website and then display some important piece of information so the public could see it. Things like stock quotes, web hits, or sales information.

In my lab, I use one to display the time. When a visitor arrives, the sign beeps and displays a message "Arrival."

You could use the interface to display real time race times or event times.

I recently started teaching my teenage daughter how to drive. I make her keep it under 40 MPH and most other drivers will tailgate, even though we are under the posted speed limit. Well, I solved the problem. If anyone gets too close, I just load the message shown in Figure 4 and you wouldn't believe how fast they back off.

I won't tell you the other messages I have loaded.

I have also created an application note showing how to connect a microcontroller to one of these signs. You can find that at the Kronos Robotics website at **www.kronosrobotics.com**

Experiment and have fun, and be sure to visit the "Control Your World" forum at **www.kronosrobotics.com/forums/viewforum.php?f=21** **NV**

Program 2

```
'BetaBrite Real Time Message Sample
func main()
  BBSetTime(1,"1215")

Loop:

   BBSendText(1,"Nuts and Volts")
   pause 1000
   BBSendText(1,"Is Real Cool")
   pause 1000

   goto Loop

endfunc


func BBSendText(tPort as integer,ttext as string)
   Const Ch1 1 'Zues Channel 1

   'Open the ComPort
   ComOpen(Ch1,baud=9600,port=tPort,parity=2,bits=7,stop=2)

'Tell the Sign we are ready to comunicate
   ComOutput Ch1,chr(0)+chr(0)+chr(0)+chr(0)+chr(0)
   ComOutput Ch1,chr(1)
   ComOutput Ch1,"Z00"

  'Send Priority Text Message
   ComOutput Ch1,chr(2)+"A0"
   ComOutput Ch1,ttext
   ComOutput Ch1,chr(4)

   ComClose Ch1
endfunc

func BBSetTime(tPort as integer,Time as string)
   Const Ch1 1 'Zues Com Channel 1

   'Open the ComPort
   ComOpen(Ch1,baud=9600,port=tPort,parity=2,bits=7,stop=2)

'Tell the Sign we are ready to comunicate
   ComOutput Ch1,chr(0)+chr(0)+chr(0)+chr(0)+chr(0)
   ComOutput Ch1,chr(1)
   ComOutput Ch1,"Z00"

  'Send Time
   ComOutput Ch1,chr(2)+"E "
   ComOutput Ch1,Time
   ComOutput Ch1,chr(4)

   ComClose Ch1
Endfunc
```

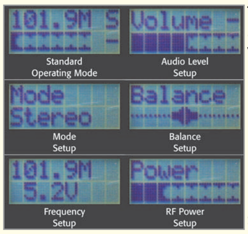
FM Broadcasters

Where YOU become the DJ!

Digital FM Stereo Transmitters

- ✔ Rock stable PLL synthesized
- ✔ Front panel digital control and display of all parameters!
- ✔ Professional metal case
- ✔ Super audio quality!
- ✔ 25mW and 1W models!

For nearly a decade we've been the leader in hobbyist FM radio transmitters. Now for 2005 we introduce our brand new FM30 series of FM Stereo Transmitters! We told our engineers we wanted a new technology transmitter that would provide FM100 series quality without the advanced mixer features. They took it as a challenge and designed not one, but TWO transmitters!



The FM30 is designed using through-hole technology and components and is available only as a do-it-yourself kit, with a 25mW output very similar to our FM25 series. Then the engineers redesigned their brand-new design using surface mount technology (SMT) for a very special factory assembled and tested FM35WT version, with 1W output for our export market! Both are designed around an RF tight vinyl clad metal enclosure for noise free and interference free operation. All settings are done through the front panel digital control and LCD display! All settings are stored in non-volatile memory for future use.

Both the FM30 and FM35WT operate on 13.8 to 16VDC and include a 15VDC plug-in power supply. The stylish metal case measures 5.55"W x 6.45"D x 1.5"H and is available in either white or black. *(Note: The end user is responsible for complying with all FCC rules & regulations within the US, or any regulations of their respective governing body. FM35BWT is for export use and can only be shipped to locations outside the continental US or valid APO/FPO addresses or valid customs brokers for end delivery outside the continental US).*

FM30B	**Digital FM Stereo Transmitter Kit, 0-25mW, Black**	**$199.95**
FM35BWT	**Digital FM Stereo Transmitter, Assembled, 1W, Black**	**$299.95**

Professional Synthesized Stereo FM Transmitter

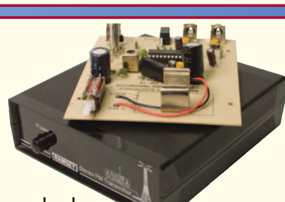
- ✔ Fully synthesized 88-108 MHz for no frequency drift
- ✔ Line level inputs and output
- ✔ All new design using SMT technology

Need professional quality features but can't justify the cost of a commercial FM exciter? The FM25B is the answer!

A cut above the rest, the FM25B features a PIC microprocessor for easy frequency programming without the need for look-up tables or complicated formulas! The transmit frequency is easily set using DIP switches; no need for tuning coils or "tweaking" to work with today's "digital" receivers. Frequency drift is a thing of the past with PLL control making your signal rock solid all the time - just like commercial stations. Kit comes complete with case set, whip antenna, 120 VAC power adapter, 1/8" Stereo to RCA patch cable, easy assembly instructions, and the SMT parts are factory preassembled - you'll be on the air quick!

FM25B	**Professional Synthesized FM Stereo Transmitter Kit**	**$139.95**

Tunable FM Stereo Transmitter



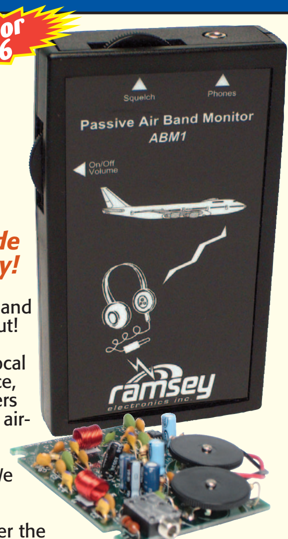
- ✔ Tunable throughout the FM band, 88-108 MHz
- ✔ Settable pre-emphasis 50 or 75 µSec for worldwide operation
- ✔ Line level inputs with RCA connectors

The FM10C has plenty of power to cover your home, back yard, or city block. You'll be amazed at the exceptional audio quality of the FM10C. Tunes through the entire 88-108MHz band. Re-broadcast your favorite music commercial free and with the dynamic range the musician intended, without all that nasty compression the big boys use to make their station sound louder than the competition. Compression produces a noticeably muddier and less dynamic sound. Runs on an internal 9VDC battery, external 5-15VDC, or our FMAC AC power adapter.

FM10C	**Tunable FM Stereo Transmitter Kit**	**$44.95**
FMAC	**110VAC Power Supply for FM10C**	**$9.95**

Passive Air Band Aircraft Monitor

New For 2006



- ✔ Monitors the entire aircraft band without tuning!
- ✔ Passive design, can be used on aircraft, no local oscillator, generates and creates no interference!
- ✔ Great for air shows
- ✔ Patented circuit and design!

Our Patented Design Has Now Made The Most Popular Kit In Our History!

For decades we have been known for our novel and creative product designs. Well, check this one out! An aircraft receiver that receives all nearby traffic without any tuning. It gets better... there is no local oscillator so it doesn't produce, and can't produce, any interference associated with all other receivers with an LO. That means you can use it onboard aircraft as a passive device! And what will you hear? The closest and strongest traffic, mainly, the one you're sitting in! How unique is this? We have a patent on it, and that says it all!

This broadband radio monitors transmissions over the entire aircraft band of 118-136 MHz. The way it works is simple. Strongest man wins! The strongest signal within the pass band of the radio will be heard. And unlike the FM capture effect, multiple aircraft signals will be heard simultaneously with the strongest one the loudest! And that means the aircraft closest to you, and the towers closest to you! All without any tuning or looking up frequencies! So, where would this come in handy?

1. ***At an air show! Just imagine listening to all the traffic as it happens***
2. ***Onboard aircraft to listen to that aircraft and associated control towers***
3. ***Private pilots to monitor ATIS and other field traffic during preflight activities (saves Hobbs time!)***
4. ***Commercial pilots to monitor ATIS and other field traffic as needed at their convenience***
5. ***General aircraft monitoring enthusiasts***

Wait, you can't use a radio receiver onboard aircraft because they contain a local oscillator that could generate interfering signals. We have you covered on that one. The ABM1 has no local oscillator, it doesn't, can't, and won't generate any RF whatsoever! That's why our patent abstract is titled "Aircraft band radio receiver which does not radiate interfering signals". It doesn't get any plainer than that!

SPECIFICATIONS

Frequency Range:	118 MHz to 136 MHz
Receiver Type:	Patented Passive Detector
IF Frequencies:	None!
Receiver Sensitivity:	Less than 2 uV for detectable audio
Audio Output:	700mW, 8-24 ohms
Headphone Jack:	3.5mm stereo phone, stereo earbuds included
External Antenna:	Headphone cord coupled
Power Requirement:	9VDC battery
Dimensions:	2.25" x 2.8" PC Board 2.5" x 4.6" x .9" Case
Weight:	4 oz. with battery

Available as a through-hole hobby kit for easy do-it-yourself assembly or a factory assembled & tested SMT version to operate right out of the box! The factory assembled version has become a favorite at air shows across the country. Just plug in the enclosed headbuds (or your favorite Walkman style headphones, and you're listening to all the activity on the ground and in the air!

Join in the excitement of monitoring your local aircraft the easy way, with our patented tune-free receiver!

ABM1	**Passive Air Band Monitor Kit**	**$89.95**
ABM1WT	**Passive Air Band Monitor, Factory Assembled & Tested**	**$159.95**

Get The Catalog!

Get the 2006 Ramsey Hobby Catalog! ***104 value packed pages*** of the neatest goodies around with over 150 new items! Order yours today on line or give us a call. Or download the PDF!

MORE than just friendly on-line ordering!

Clearance Specials, Ramsey Museum, User Forums, Dealer Information, FAQ's, FCC Info, Kit Building Guides, Downloads, Live Weather, Live Webcams, and much more!

www.ramseykits.com

800-446-2295

ramsey
www.ramseykits.com

590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560

Where Electronics Is Always FUN!

- ✦ ***Build It!***
- ✦ ***Learn It!***
- ✦ ***Achieve It!***
- ✦ ***Enjoy It!***

For several years now, I have been wanting to design and build a period counter with totalizing function.

I finally made time to do just that. My initial design was quite basic, but then I started the what-ifs, and the original design grew to twice its size.

A WIDE RANGE PERIOD COUNTER/TOTALIZER

When I finished the first prototype, I sat back and studied it. Then I asked myself what would I want to strip to make the design simpler. The answer was, "nothing." It was very functional the way it stood, and it's a dandy.

The pleasure and convenience of use in the coming years far outweigh the few extra hours needed in construction. So there: I had what I wanted — another "mighty weapon" to add to my arsenal of bench test equipment. A simplification in design, with a slight loss in overall accuracy, is included near the end of this article for those who prefer to have less wiring to do.

You may be thinking, "Why build a period counter when I already own a frequency counter?" For starters, there are many situations where it's just more convenient to count electrical signals and read out the actual time (µs, ms, s) rather than frequency. Some frequency counters will also read out the period, but almost all of them fall short for slow repetition rates or long periods between uses. Their lower limit is around 10 Hz.

■ Front view of the counter.

The counter described here also has additional important features. In addition to its wide range of counts (from one to one billion), it will also capture and display pulse width, either positive or negative. At its extreme range, it will capture and display a single pulse as narrow as 1 µs, even if it only occurred once a day — try doing that with your oscilloscope!

On the other end of its range, it will count rates as slow as 1,000 seconds or as fast as high-speed pulse trains from a function generator's burst mode of operation. And finally, a totalizing function was incorporated into it to count electrical events and sum them (up to 100,000).

Although operation of counters are similar, they differ in several respects. A frequency counter counts an unknown clock rate of input cycles in a known gate time, usually one second, and displays the count in Hz/sec.

A period counter counts a known clock rate in an unknown gate time (input signal) and displays this as the actual time of occurrence. For this reason, period counters are a little more complex.

This counter will perform the following:

- Period count — One microsecond to 1,000 seconds
- Pulse width — One microsecond to 1,000 seconds

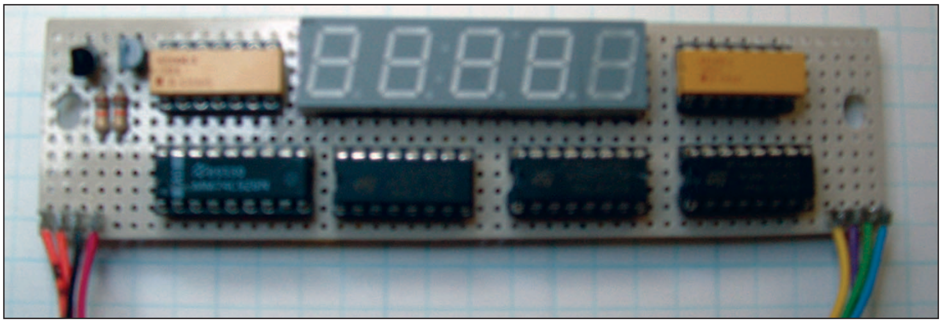
• Totalize — One to 100,000 events

Although measuring pulse width will lose accuracy below one microsecond due to resolution, it can capture and display glitches down to 100 nanoseconds. Totalizing input pulses can be as short as one microsecond or as long as a month. All of the above functions will accept any type of waveform.

Before we get into construction, I will give a complete theory of operation for understanding circuit operation and eventual troubleshooting, if necessary, upon completion. To aid in this discussion, I am including a timing diagram which will be referred to in the schematic diagram. This will be of use because there is a lot of high-speed sequential edge triggering required in this unit.

The signal under test enters J1 and is amplified by Q1. R1 limits Q1 base current and C1 aids in high-frequency response. D1 clips negative peaks to protect Q1. This stage has an overall gain of approximately three or four. Its input sensitivity is 1.4 volts peak, which is adequate for all logic families in use today except ECL (0.8 vpp), but that's beyond the intended use for this design.

The input impedance is 22 kΩ, shunted by less than 50 pF. This circuit produced no noticeable degradation to rise or fall times of the input signal. The input can be of any waveform from 1.4 vp to 50 vp (100 vvp-p). Beyond 50V, you will need an external attenuator. This can be as simple as a resistance divider if extreme speed is not an issue, and it usually isn't at these high levels.

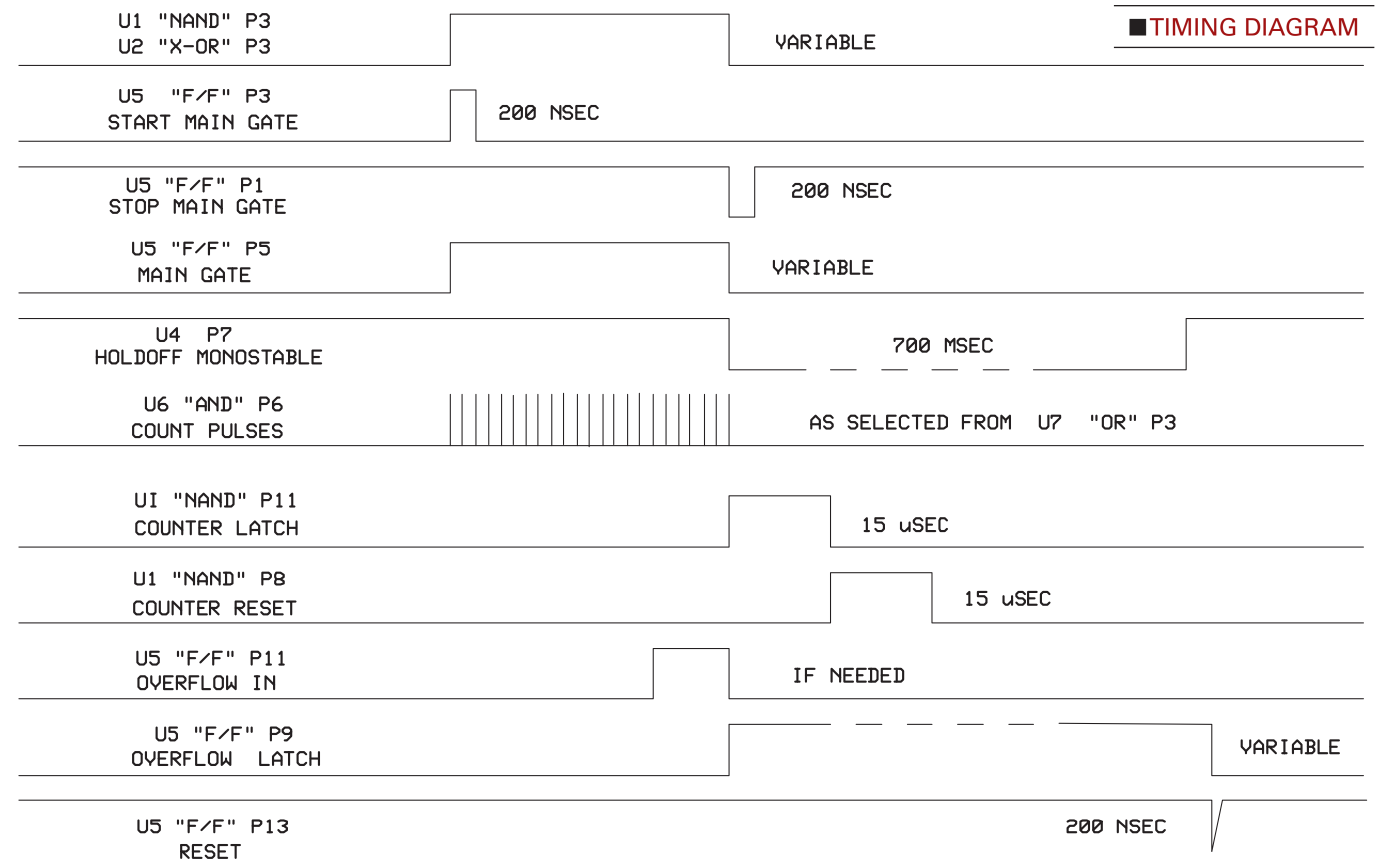


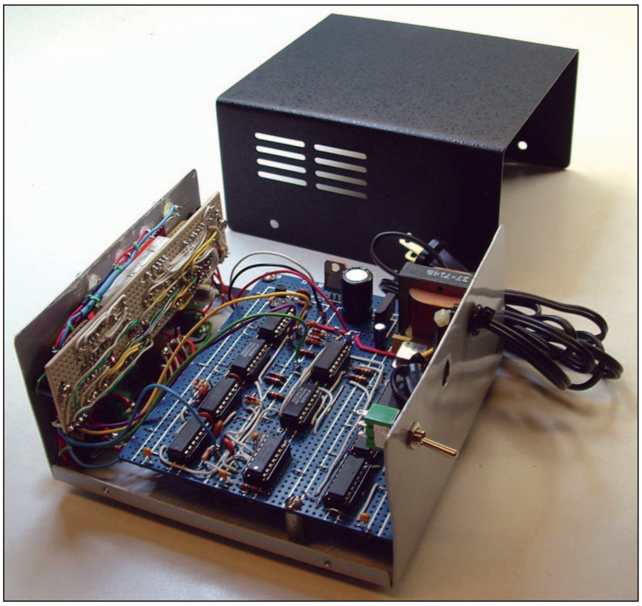
■The display board.

The output of Q1 drives U1A — a NAND gate with Schmidt-trigger action. The output of U1A is a textbook-perfect rectangular wave, no matter what wave shape or amplitude is input to J1. Only its length will vary, according to the input signal. For this reason, I started the timing diagram at this point.

The next stage — U2A — allows for selection of positive or negative edge triggering of the input signal while still

■TIMING DIAGRAM





■ A look at the internal wiring.

maintaining a positive-going pulse as its output, since this is what the circuitry that follows it wants to see. The use of an exclusive OR gate fit the bill perfectly here. A lookup of its simple truth table will bear this out.

The output of U2A is split into two paths: one to start the main gate, F/F, and one to reset the same. Assume for the moment that S2 is switched to period mode of measurement; U3B will block any signals in that path by grounding one of its inputs. The leading edge of the pulse applied to U3A is highly differentiated by C2, R4. This will produce a 200 nanosecond positive pulse to U5A's clock input (P3). Only the leading rising edges of the input will be seen by this circuit, due to the gate configuration of U3B. U5A is a D type F/F that is wired for toggle operation by connecting P6 to the data input P2.

Now, starting from its reset position, Q (P5) will go high with the first rising edge of the input pulse, low with the next rising edge, and so on. This produces a main gate pulse coincident with the period of the signal under test.

Now, assume S2 is switched to pulse measurement. U5A starts the main gate action — output goes high coincident with the rising edge of the input signal. But now we have to enable U3B by putting a high on one of its inputs. U3C will now perform exactly as the above U3A did, but with one exception — it will output a 200 nanosecond negative pulse only when the input signal is on its falling, trailing edge.

When U5A sees this negative spike at P1, it immediately resets its output to a low, and thereby aborts its toggle operation. Now the output at this point is an exact replica of the input pulse width. Up to this point at U5A P5, we have an exact gating pulse based on either period or pulse width and triggered from either positive or negative edges.

Also from this point the gating pulse diverges in three directions:

1. It enables main gate U6A for its duration and allows selected master clock pulses to pass through to the counting circuits of U11A and U13.

2. It drives U7A to deliver the necessary timing pulses for latching and resetting of the counters. U7A is a leftover OR gate from a quad IC, and is put to good use here as a buffer to drive the high input capacitance (1,000 pF) of the U1B circuit. Without this buffer, some deterioration of the main gate signal would occur.

3. Finally, it triggers U4A P5, which is a dual monostable IC. Its role is that of a holdoff circuit. As soon as we complete one cycle of the main gate pulse, the negative edge triggers P5 and produces a low at the P7 output which disables U3A so that U5A cannot operate again until the monostable times out (700 ms). This action immediately blocks any input pulses from entering U5A's clock input and keeps it that way. When the holdoff times out, U3A is once again enabled and will operate on the next incoming signal pulse, at which time the whole operation will repeat itself. The reason for this holdoff circuit is to latch the count circuit into the display long enough to read it. Without it, the digits would flicker constantly at higher period rates. The holdoff time of 700 ms is arbitrary and may be changed by adjusting the time constant of C5, R7.

At this point, we will switch from analyzing signal flow through the front end and focus on the time base. The accuracy of any unit is that of its time base and is given as:

Time base error in PPM,
± resolution, ± one display count.

I decided to use a packaged oscillator (eight-pin DIP) as it does not cost much more than a lone crystal — $1.70 for this unit. The manufacturer guarantees ±100 PPM, but of the several units I purchased, they were within 5 PPM at room temperature. This accuracy is far beyond what the display can resolve, almost removing it from the equation.

Since we cannot escape the ±1 digit in the display, accuracy depends almost totally on resolution. The lower the display count, the greater the possible error. For example, with a display of 10, due to resolution error, the actual count could be closer to 9 or 11, a possible error of 10 percent. This would be an extreme case. The higher the count, the greater the resolution, hence the greater the accuracy. The only way to get around this is to add more digits to the display. I stopped at five digits, because I felt I was at a point of diminishing returns. I have checked this unit against an expensive laboratory counter at full display, and it was right on the money!

With that said, we will continue with time base and master clock operation. XO is a 10 MHz oscillator (100 ns), which is divided by 100 (10 µs), and by 100,000 (10 ms) through the divider chain of U8, U9, U10. Three clock rates can be input to U7bcd (10 MHz, 100 kHz, 100 Hz). The fourth input is 5 VDC for totalizing. Selection of these inputs is accomplished by S4. Starting at the top A position, 5 VDC is applied to

U6B gate, enabling the 10 MHz clock (100 ns) to pass through U7 to the input of U6A. Also Q2's collector is energized to light the correct decimal point. At the same time, a front panel LED is lit through R14. This would be labeled *usec* and its count range will be 0-9999.9 µs.

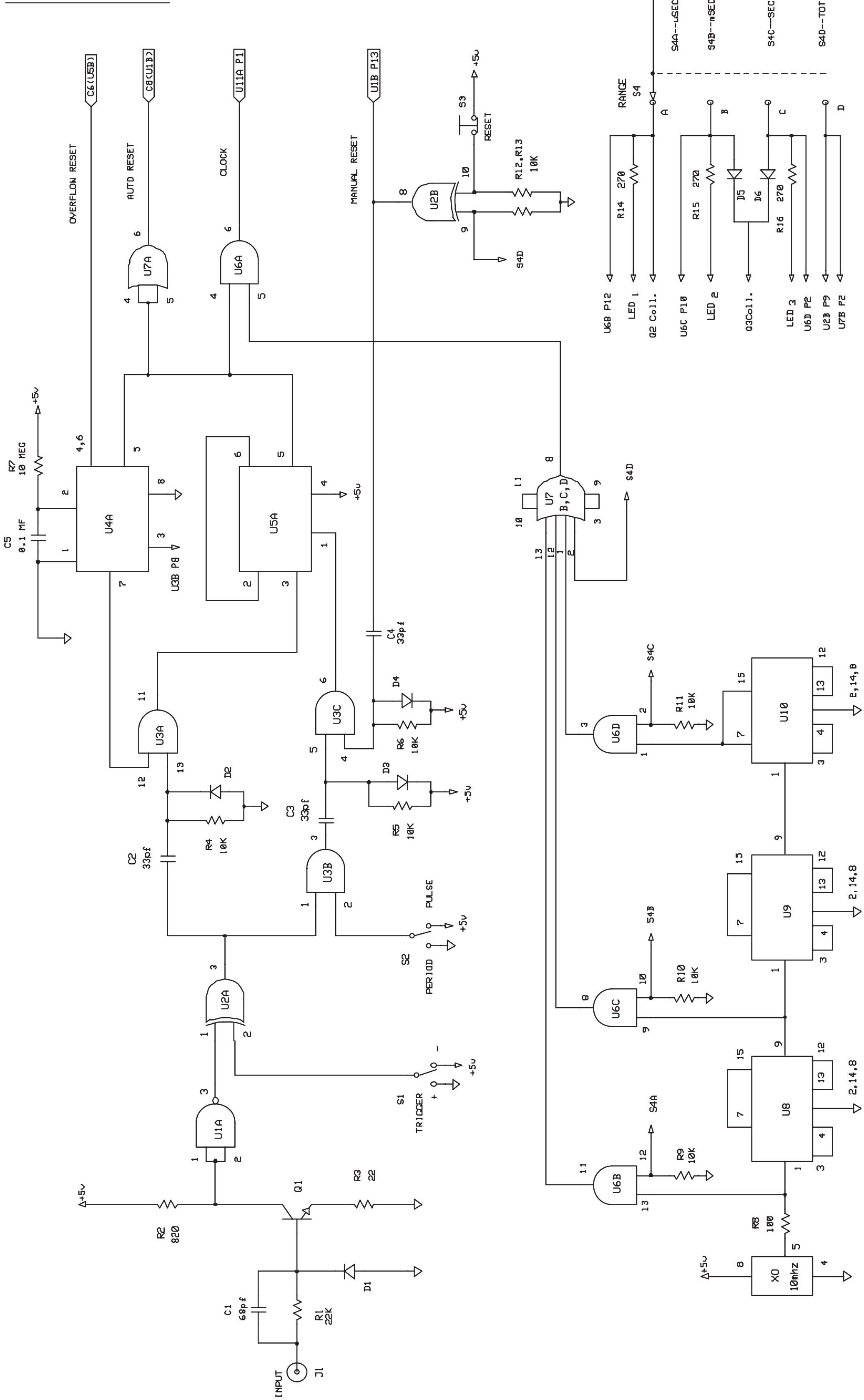
With S4 switched to B position, the U6c gate is enabled, passing the 100,000 kHz (10 µs) clock rate to U6A. Again, R15 lights the appropriate front panel LED. Also Q3's collector is energized, shifting the decimal point. This is labeled *msec* and its count range is 0-999.99 ms.

With S4 in position C, U6D is enabled, passing the 100 Hz (10 ms) clock rate U6A. As before, R16 lights the appropriate front panel LED. The same decimal point is lit as in B position. Diodes D5 and D6 isolate these switch positions from inadvertently being energized from backfeeding. This range is labeled *sec* and its count range is 0-999.99 s.

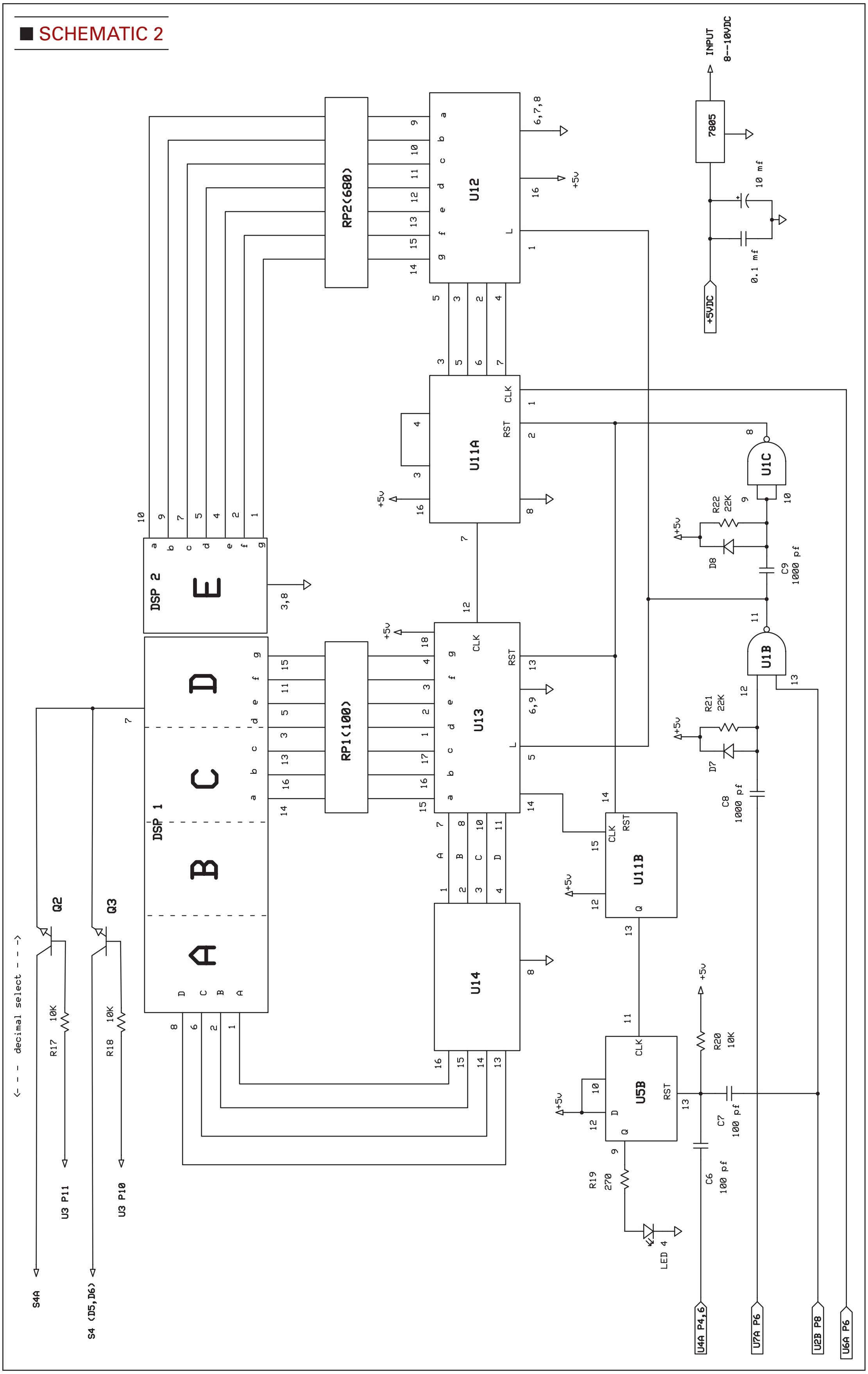
Finally, with S4 in position D, all clocks are disabled, and a steady high is applied to U6A. Now we can count the actual gate pulse as one digit per pulse for a totalizing function. Also, the manual rest line is pulled low and kept there, thereby enabling the counters to count and display continuously without resetting and allowing totalizing to occur. One word of caution here — S2 must be in pulse mode when totalizing.

Now we are back to where we left off on the front-end circuitry at U6A. When U6A P4 goes high, the clock rate we have

■ SCHEMATIC 1



■ SCHEMATIC 2



selected will be present at U6A P5 input and passed through to the first counter U11A P1, which is a high-speed decade counter. Its overflow goes to U13, a multiplexed four-digit counter, latch, decoder. When main gate U6A P4 goes low, all counting stops and is stored in the counters. At this point, we have counted a known clock rate in an unknown gate time. When the count state is latched into the display, it will read out the gate time (our input signal under test) in real time.

For example, let's say we had selected the S4A µs range and had counted and latched 5,000 clock pulses (10 MHz; 0.1 µs) during the time the main gate was enabled. The display would read 500.0 µs (5,000 times 0.1 µs). This is the period or pulse width from the circuit under test. Backing up just a bit, when U6A P4 went low, the first event to happen was that the counters stopped counting. The same main gate pulse was also coupled through U7A to U1B P12. U1B is a simple one-shot circuit, whose output is dependent on the RC time constant of R21, C8. In this case, it outputs a 15 µs positive pulse when triggered from the trailing negative edge of the main gate pulse. This pulse latches the count state into the display.

Also, the trailing negative edge of this

PARTS LIST

ITEM	DESCRIPTION	ITEM	DESCRIPTION
❑ R1	22 kΩ	❑ U1	74HC132 quad NAND gate
❑ R2	820 Ω	❑ U2	74HC86 quad exclusive OR gate
❑ R3	22 kΩ	❑ U3,6	74HCo8 quad AND gate
❑ R4, 5,6	10 kΩ	❑ U4	74HC4538 dual monostable
❑ R7	10 MΩ	❑ U5	74HC74 dual D flipflop
❑ R8	100 Ω	❑ U7	74HC32 quad OR gate
❑ R9, 10,11,12,13	10 kΩ	❑ U8,9,10,11	74HC390 dual decade counter
❑ R14,15,16,19	270 Ω	❑ U12	CD14543 four-digit display ctr,dec,drv (Jameco)
❑ R17,18,20	10 kΩ	❑ U14	ULN2003 npn inverters
❑ R21,22	22 kΩ	❑ Xo	10 MHz ttl osc. (Mouser 520-TCH 1000)
		❑ Xo	1 MHz ttl osc. (Mouser 520-TCH 100)
❑ C1	68 pF	❑ S1,S2	SPDT toggle switch
❑ C2,3,4	33 pF	❑ S3	Pushbutton switch NC
❑ C5	0.1 mF	❑ S4	Rotary switch – four position
❑ C6,7	100 pF	❑ RP1	14 DIP resistor, 100 Ω
❑ C8,9	1000 pF	❑ RP1	14 DIP resistor, 680 Ω
❑ D1 thru D8	IN914	❑ Display #1	Four digit, 0.4 inch, multiplexed
			(Digi-Key 160-1551-5-ND)
❑ LED1,2,3	10 mA red	❑ Display #2	One digit, 0.4 inch (Digi-Key 160-1533-5-ND)
❑ LED4	10 mA yellow	❑ Chassis	4OUB103 EPD (Mouser)
❑ Q1,2,3	2N3904		

pulse triggers an identical circuit, U1C. This is the auto reset circuit. The output of this stage resets the counters to zero and readies them for the next measurement. For manual reset, this line is normally held high. When S3 is depressed, the line goes low, resets the auto reset circuit, and directly clears the main gate F/F through U3C, the holdoff mono. It also clears U5B (overflow latch) through C7. This same latch will also be cleared automatically when the half circuit times out through C6.

As mentioned before, while in a totalizing operation, a constant low is put on the manual reset line by S4 D action. This is needed to continuously latch through the count state from the counter. This presented a problem for manual reset because the reset line was already low! Once again, the exclusive OR gate, U2B, came to the rescue, allowing manual reset to occur regardless of which state that line is in.

There is just one more topic to cover before construction — the overflow circuitry. Without an overflow indicator, ambiguity becomes a problem. When U13 completely loads up the counters and goes into overflow condition, its carryout (P14) sends a signal to U11B, which acts as a sort of decoder. The output of this stage operates a latch {U5B) to light the overflow LED from P9.

One other item that may have piqued your curiosity is the circuit associated with the decimal points. These points on the display are multiplexed with no provision to light them externally. This was quite annoying, and for that reason a decoder had to be added (Q2 and Q3) to light them at the proper time.

This about covers the theory of operation; now onto construction. I built the display circuitry on a 1-1/4″ x 5″ perforated board. This board contains both displays, U11, U12, U13, U14, RP1, and RP2. I mounted the board behind the front panel on 3/8″ standoffs such that it lined up to a 1/2″ x 2-1/2″ display window cut into the panel.

The rest of the circuitry, including an AC power supply, was mounted on a 5″ x 4-1/4″ perforated board, as

shown in the photo on page 48. Everything was installed in the chassis shown in the Parts List. It was a tight fit, and I might opt for the next larger size in this series were I to construct another one. I did not detail a power supply in the schematic, since most constructors use what they have on hand from their junk box.

The properly bypassed, 5V regulator (7805) shown in the print can be driven from any DC source of 7.5-12 VDC. I show the power supply components that I used in the photo on page 48. This was a small 120/7.5 VAC transformer, full-wave bridged into a 1,000 µF 'lytic to provide an 8V source for the 7805. Power requirements are 5 VDC at 160 mA with all digits lit to 8.

To conserve space, you could use a wallwart type of transformer. Or, if you prefer battery operation, use four AA cells in series with an IN4001 diode. This will give a supply voltage of about 5.5 VDC; no 7805 used here.

Also, for battery operation, you can replace RP1 with a 330 Ω resistor and RP2 with 2,000 Ω. This will cut the current consumption in half (50 mA average) and still give a decent readout. This sort of readout-brightness/current-draw tradeoff comes down to preference, and is thus best determined by the user.

One last item about construction — as mentioned earlier in this text, you can simplify this project (with some loss of overall accuracy) by eliminating U11, U12, RP2, display No. 2, Q2, and Q3. You will end up with a four-digit counter that has no decimal point. The clock input U11A P12 then goes directly to U13 P12. Change the 10 MHz XO to 1 MHz XO, which is shown in the Parts List, and eliminate all front panel LEDs.

The ranges will now be:

- S4A 0-9,999 microseconds
- S4B 0-9,999 microseconds
- S4C 0-9,999 seconds
- S4D Totalize to 9,999 counts

This will require a little more care in use. Without an overflow light, you may have to work down from a slower range to eliminate ambiguity. This modification will also reduce current consumption for battery use. You will also need to change the clock output connections to accommodate these ranges:

- Microseconds — U8 P1 (same)
- Milliseconds — U9 P7
- Seconds — U10 P9
- Totalize — Same

In addition to being a very useful piece of test equipment, this is an interesting and educational project, since it involves many types of logic applications — display circuitry, high-speed sequential timing, etc.

Be sure to ground all unused IC inputs. Take your time wiring it, and double-check your connections (the most common source of errors) as you proceed, and you should have no problems. **NV**

ETCH A SKETCH® — BIG TIME

The world's largest Etch A Sketch made its debut at SIGGRAPH 2006 — the 33rd International Conference and Exhibition on Computer Graphics and Interactive Techniques held July 30-August 3, 2006 in Boston, MA.

Officially endorsed by The Ohio Art Company, this interactive installation enabled audience members to control (in real time) the two main drawing knobs that are found on a conventional, err, analog Etch A Sketch. The audience's collaborative drawing was displayed on the main projection screen. Audience members were also able to "shake" the screen and erase the screen's contents.

COMING SOON TO YOU!

Another feature at SIGGRAPH 2006 was the Emerging Technologies program. Featuring a broad range of installations from research labs, universities, and independents, this program gives attendees a quick "heads up" for the types of human/computer interaction that could be invading your personal space in the coming years.

Selected from 110 submissions from 18 countries, only 36 technologies were on display. The following are just a few highlights of this popular program:

- Virtual Open Heart Surgery: Training Complex Surgical Procedures in Congenital Heart Disease. Contact: Thomas Sangild Sorensen, University of Aarhus, Denmark
- Forehead Retina System. Contact: Hiroyuki Kajimoto, The University of Tokyo
- Powered Shoes. Contact: Hiroo Iwata, The University of Tsukuba
- Perceptual Attraction Force: The Sixth Force. Contact: Tomohiro Amemiya, NTT Communication Science Laboratories
- The Virtual Humanoid. Contact: Michihiko Shoji, NTT DoCoMo
- The Huggable: A Therapeutic Robotic Companion for Relational, Affective Touch. Contact: Walter Dan Stiehl, Massachusetts Institute of Technology Media Lab, Robotic Life Group

YOU'RE GONNA FLY WHAT AND YOU'RE GONNA FLY WHERE?

The Space Shuttle Discovery landed with the crew of STS-121 at the NASA Kennedy Space Center, FL. Photo by NASA/Bill Ingalls and courtesy of NASA.

Every time the space shuttle returns from a successful mission, NASA is quick to promote some of the exotic research that was performed during the quick jaunt into space. So, there wasn't too much surprise when NASA reported that Space Shuttle Discovery carried a cargo of 150 fruit flies during its July 2006 mission. What did raise a couple of thousand eyebrows around the world was learning that these annoying little flies were being studied for the effects of long-term space flights on the human immune system.

If my college zoology courses were worth a darn, fruit flies belong to the genus Drosophila. Likewise, these diminutive flies are a researcher's dream come true — they reproduce quickly, have a well-documented genetic makeup, and they have around a 60 day life cycle. So, every two months you can study a new generation of fruit flies.

And that's where studying the Drosophila immune system comes in. In this case, the fruit flies from Discovery will be infected with some fungus and bacteria. Similarly, a "control group" of Earthbound fruit flies will be infected with the same fungal/bacterial cocktail. Scientists will then study the immune systems from the two fly groups. Hopefully, the answers gleaned from these experiments will help to contribute to making long endurance space flights lasting more than two years a piece of cake for future astronauts.

So today fruit flies — tomorrow, the next stop Mars.

UP THE CREEK WITHOUT A PHOTOVOLTAIC CELL

Did you hear that? That was the sound of a solar-powered boat sailing by you. Puttering around Serpentine Lake in London's Hyde Park is a unique boat that is totally silent and completely pollution free. Known as the Serpentine SolarShuttle, this boat is powered by a beautifully arranged overhead netting of photovoltaic cells.

Designed by the incomparable Christoph Behling, Serpentine SolarShuttle is built by SolarLab Research & Design; a research and design firm founded by Behling. There are other solar-powered transportation boats in the SolarLab fleet, including the world's largest solar-powered boat, the Hamburg SolarShuttle.

If you'd like to read more about the Serpentine SolarShuttle and SolarLab, go to **www.solarlab.org**.

TI WANTS YOU TO WIN A DLP HDTV THE EZ WAY

Get those thinking caps on; Texas Instruments has just announced an MSP430 microcontroller (MCU) eZ design contest. Running from July 17 through October 9, 2006, contestants may submit projects ranging from practical everyday devices to industry-specific solutions. Regardless of your inspiration, your entry must show off the world's lowest-power MSP430 MCUs and development tools, including the new eZ430-F2013 — a USB stick based full emulation and development tool.

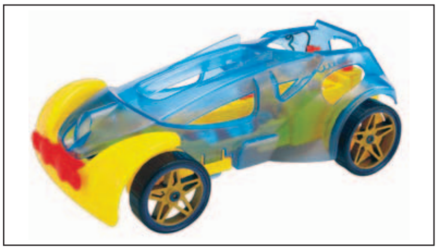
Only original product designs should be submitted, along with all of the usual suspects: design schematics, firmware, tech diagrams, photographs, and videos of running hardware. The main processor for all design entries must be the MSP430 MCU.

Judging will spotlight innovation and the best employment of on-chip MSP430 MCU features. The grand, first, and second place winners will

receive a 61″, 50″, and 42″ Samsung DLP HDTV, respectively, along with free entry and lodging for the 5th Annual MSP430 Advanced Technical Conference (ATC) held in Dallas, TX on November 7-9, 2006.

For additional instructions on how to enter the MSP430 MCU eZ Design Contest, eligibility, design selection process, and notification, please go to **www.ti.com/designmsp430**.

KNIGHT INVADER III WINS TOY OF THE YEAR AWARD



The newly dubbed "Toy of the Year" — Knight Invader III.

The *Creative Child Magazine* awards panel has presented Knight Invader III from OWI, Inc., with "Toy of the Year Award" in the electronic educational toys for kids category.

The future of toys has arrived. Knight Invader III is a super race car that includes mini-construction projects, teaches the basic principles of electricity, gear mechanisms, and demonstrates two different modes of propulsion.

This sleek super racer has it all and a little bit more. Knight Invader III can propel itself off the starting line and challenges any contender to try to keep up when it is set into direct motor drive. When on a mission, Knight Invader III can secretly change its transportation mode to solar power (sold separately). Plus, for the educationally minded guardians, this fire plug kit includes mini construction projects that teach the basic principles of electricity including experiments involving series and parallel circuits and solar energy (sold separately). Another "bright" idea by Dr. OWIKIT is the inclusion of a bulb and switch that allows users to investigate simple experiments in conductivity. No tools required for assembly. **NV**


1,300+ Products for the Hobbyist, Inventor, & Engineer
Free Forums and Resources
www.DesignNotes.com
Plastic & Aluminum Project Cases
As Low As $5.80
Connectors & Adapters
As Low As $0.40
100's of Kit Projects
Remote Control By Telephone Only $65.00
K6501
Test Equipment
USB PC 2 CH 60MHZ Oscilloscope
New Product
Includes 2 Probes! $334.95
PCSU1000
Free with orders over $100.00
1 year Subscription to NUTS AND VOLTS (Print Edition)
or
DVM850BL Digital Multimeter
DesignNotes.com
Affordable Prototyping Products
Toll Free
1-800-957-6867
15 Day Money Back Guarantee
DesignNotes.com, Inc
2014 Ford Road, Unit B • Bristol, PA 19007


Add SD in a FLASH!
with our
Flash Card Library
Easily exchange data between your product and a PC
Create, Append and Delete PC compatible files
Add GIGABYTEs of onboard or removable storage to your product
Compatible with over 20 card styles SD, MMC, MicroSD, TransFlash, …
No OS, RTOS or interrupts are needed
Works with low cost Microcontrollers
Support for FAT16 and FAT32 file systems
Simple API calls with full C sources
SD/MMC
Serial Flash Card Library
To get you up and running quickly, Development Boards and Adaptors are available for a variety of Micros.
Get More Details at www.ecs87.com/fclib
EFFICIENT COMPUTER SYSTEMS, LLC
5 Emerson Drive Center Barnstead, NH 03225 • 603-776-3151
Offering MEGA solutions for the embedded MICRO world.
ECS87.COM
Introductory Prices starting at $139.95


CHINA PCB SUPPLIER
DIRECT SALE FROM CHINA
PROTOTYPE TO PRODUCTION
instant online quote
shopping cart ordering system
China competitive prices
free Electrically test
web: http://www.pcbcart.com
email: sales@pcbcart.com
Tel: +86-571-87013819
Fax:+86-571-87036705
Add: No.2 Huixing Road, Hangzhou, China
WWW.PCBCART.COM


You need these cutting edge products to quickly solve your toughest design and measurement problems!
New Products
Dualcore....dual ARM IC
Emulin....LINbus/J2602 emulator
WIZnet....TCP/IP IC
Hunt RTG....instant FPGA/DSP
Byteparadigm....USB 2.0 I/O
Systembase....Quad UART chip
Owasys....embedded wireless
RH-02 / PT104
RH-02 - Temperature and Humidity Logger for PC. PT104 - High resolution and accuracy. Platinum sensors. USB powered - no power supply required. Free s/ware.
from $379/759!
Touch-sensing ICs
Quantum ICs- World-beating capacitive sensor ICs for switching and control with patented features like Adjacent Key Suppression, Spreadspectrum, adjustable sensitivity. QRG ICs in use worldwide by the world's largest consumer/appliance manufacturers
<$1 (10K)
Universal Meter
Unimeter - User-configurable data logger/controller/meter with built-in real-time math calculations. 100s of uses: process control, logger, 5 digit LCD, bargraph, A/D, I/O.
Unimeter $240
USB to 24 x I/O
USBI/O24 - 24 latched I/O lines self-powered from your PC's USB port. Ubicom SX52 allows individual pin I/O programming. I/O control via PCs USB port.
from $69!
USB-inSync Logger
FATfile Storage
Touchpanel Controller
CAN-USB
Austronics AEEMA '2005 Best New Product' award!
Cleverscope CS328
PC oscilloscope, logic analyzer, and Signal Generator all-in-one!
Cleverscope CS328 - NEW 100MHz USB-connected mixed-signal 2-ch PC scope with 8 digital inputs and sig.gen. Huge 4MSample buffer for deep data drilling and features exceeding most expensive stand-alone scopes (sophisticated analog/digital signature/ pulsewidth triggering, glitch capture, whole buffer filestore, "copy/paste" results, etc.). Find hard-to-isolate waveforms! Blows other scope adapters away! Display math-modified and digital signals together. Join multiple units for more channels!
"CS328 is absolutely wonderful, I'm thrilled to bits with it. The software is light-years in front of that (other PC scope). The Math function makes me almost believe I'm using a LxCxxx." Roger – EFE Ltd
• 2 x 10-bit, self-calibrating A/D channels - 100 MSa/s simultaneously!
• 8 digital inputs - 100 MSa/s simultaneously!
• Huge 4M Samples storage per channel – drill deep!
• Simple "Copy and Paste" into other applications such as Word, Excel
• Highly flexible mixed-signal triggering, scope and spectrum display.
• View small ac signals on large ac or dc offset
• Built-in math capability - multiple display: A-B, invert, diff/integrals, etc.
• Includes all probes cables power supply, s/w, manuals.
• Optional Signal/sweep generator CS700 - +$199.00 Sine, Square: 0.2Hz–10MHz; Triangle: 0.2Hz – 1MHz - Frequency Shift, Phase Shift diff., phase/gain - great for filter, amplifier testing!
CS328 - usually $1045 - only $999 !
• Scaling and offsetting to view 50mV full scale offset to any value between –8 and +8 V.
• Gain automatically set from 20mV to 800V full scale.
• Offset automatically set from 0 to ±4 or 40V in 10/100 mV increments (e.g. 20 mV super-imposed on 3V DC).
• Analog triggering to 1% of display ht, with LP, HP or noise filter.
• Ext. trigger, adjustable from 0 to ±20V in 40 mV increments.
• Simultaneous 8 x digital inputs sampling at 100 MSa/s (threshold 0 to 8 V; 10 mV increments)
• Hardware trigger on rising or falling edge on any input signal, digital input combination and a min or max duration (pulse width).
• Self calibration to ensure DC performance specifications.
Cleverscope software resources include:
• Separate, freely moveable and resizeable windows to display the signal, a zoomed signal view, and the frequency spectrum of the signal, and control panel.
• Spectrum analysis with conditioning windows - log or linear.
• Signal averaging (exponential, block and peak hold) and LPF.
• Math calc including + - / * sqrt, integral, differential, and filtering.
• Signal msmt, including Pk-Pk, RMS, DC, pulse width, T and F.
• Copy and Paste graphic or data to other applications.
• Save and Open from disk.
• User defined units, signal names and scaling (offset and gain).
• Text annotation of each graph.
• Web server for remote viewing of LAN connected unit.
TR-72W $527!
with free software.
PP298 $1269.95!
etc. Optional sophisticated software allows customizability.
DLP-D only $12.99!
powered USB-based digital temperature/humidity sensor. Easily monitor temp/humidity /dew.pt. from PC's USB port.
DLP-TH1 from $99.95!
GPS Logger
TrackStick™ - gumpack-sized logger tracks position data for shipments, vehicles, personnel, etc. Records date, time, location, speed, heading and altitude at pre-set intervals - shows route/speed on PC! $249.00
USB Logic Analyzer
ANT8 Matchbox-sized USB-powered logic analyzer. Sample 8 channels at up to 500 million samples-per-sec. View traces on PC. Print or save for later review! (16-ch. ver. too!) $222!
EMI Tents
EMI shielding tents made from Ni/Ag fabric. -86dB 20MHz to 18GHz. Portable - tabletop to room/lab size for on-site, on-demand pre-compliance testing. External PVC or aluminum frame. From $3500
USB Bus Analyzers
Tracker 110/Explorer 200 high-perf. economical USB 1.1/2.0 protocol analyzers. $999/$2999! USBEX300 for debugging USB wireless Wimedia <$29,500! New - Bluetooth Sniffer <$10K
USB-Serial Adapters
Serial port for laptops without one! USB-COM instantly updates older RS232 products to USB! PC thinks it's using COMport up to 8 channel. only $35 (1) or $25 (100)
microUSB - Tiny USB ready-made USB Interface to 1 Mbps 0.7" x 0.5" based on SiLabs CP2101. Gnd/RX/TX Suspend, Reset, & 3.3V 100mA source –40°C +85°C.
microUSB only $25!
I2C for PCs
PCI93LV: industry-standard I2C card for PCs. WINI2C/PCI software gives windows-interface to develop and debug I2C bus systems. UCA93LV is new USB version. NOW! - transparently monitor at 400kHz! only $499!
FrontDesign
FrontDesign - software for designing professional-looking front panels - with a host of drawing functions. Print, mill or engrave, or export final at 600 dpi.
FrontDesign only $79.95!
Ether-IO
Ether-IO - UDP/IP-controlled 24 X digital I/O board. 3 x 8-bit TTL ports each independently programmable. Connects to any TCP/IP Ethernet network.
Ether-IO 24 from $99!
Signal Wizard
Signal Wizard - easy-use real-time DSP-based filter board for audio bandwidth signals. Design filters in seconds without any DSP knowledge!
Signal Wizard II only $399!
USB to I2C
USB I2C/IO - provides a simple "drop-in" solution for connecting your PC to 90kHz I2C + 20 I/O lines. Free software. Use multiple boards for more I2C/IO.
USB I2C/IO only $79!
FREE COFFEE
Call 1-888-7SAELIG
Mention Offer# SBW
to get a free STARBUCKS Card with your order!
While supplies last - not available with any other offers
Saelig Co. searches the world to bring you unique, easy-to-use control and instrumentation products from overseas. Customers include: Intel, Philips, NEC, Kodak, Nokia, US Military, Microsoft, Dell, Xerox, Universities, T.I., Harris, Sony, J&J, Thomson, Sandisk, General Dynamics, H-P/Compaq, etc.
Saelig.com unique electronics
888-7SAELIG info@saelig.com
Also: Wireless Temp Loggers, Stepper Motor Controller, PC for harsh environments, Automotive Diagnostic Kit, PC for Healthcare, Wireless controller, Oscilloscope, JTAG testing, Indicator modules, Video Motion Detectors, Industrial IO modules, Software for soundcard - front panels - logic simulation and hardware control - CAD
If you don't see what you need maybe we can find it for you? - Ask for sales!

UNDERSTANDING

Digital Logic ICs

PART 3 — TTL Basic Usage Rules

Ray Marston takes a further look at the "74-series" of digital ICs and at basic TTL usage rules in this installment of a four-part mini-series.

BY RAY MARSTON

It is usually a fairly simple matter to design logic circuitry using "74-series" TTL ICs, provided that a set of TTL basic usage rules are observed. Assuming that the matter of fan-in and fan-out has already been taken care of (as described in last month's Part 2 of this mini-series), four other basic usage themes remain, and these are described in the next few pages under the headings *Power Supplies*, *Input Signals*, *Unused Inputs*, and *Interfacing*.

Power Supplies

The 74-series TTL ICs are designed to be used over a very limited supply voltage range (4.75-5.25 V), and — because they generate very fast pulse edges and have relatively low noise-margin values — must be used with supplies with very low output impedance values (typically less than 0.1 Ω). Consequently, practical TTL circuits should always be powered from a low-impedance, well-regulated supply such as one of those shown in Figures 1 to 3, and must be used with a PCB (printed circuit board) that is very carefully designed to give excellent high-frequency supply decoupling to each TTL IC.

In general, the TTL circuit's PCB's +5 V and 0 V supply rail tracks must be as wide as possible (ideally, the 0 V track should take the form of a ground plane). Connections and interconnections should be as short and direct as possible. The PCB's supply rails should be liberally sprinkled with 4.7 µF tantalum electrolytic capacitors (at least one for every 10 ICs) to enhance low-frequency decoupling, and with 10 nF disk ceramics (at least one for every four ICs, fitted as close as possible between an IC's supply pins) to enhance high-frequency decoupling.

Three power-supply circuits are shown in Figures 1 to 3. These are typical TTL PSU designs, and all work basically the same way. In each case, the AC input voltage is stepped down to a value in the 9-12 V range via transformer T1 (which has a VA power rating at least double that of the final power supply's DC output). The resulting T1 output is full-wave rectified via bridge rectifier BR1 and converted into a reasonably smooth DC voltage via electrolytic capacitor C1.

The DC output of C1 is then converted into a smooth and stable 5 VDC via a 7805-type 5 V voltage regulator IC1 (which requires an input voltage that is at least 3 V greater than the specified output voltage). In Figure 1, IC1 has a 100 mA current rating, as implied by the L in the middle of the IC's type code. In Figures 2 and 3, IC1 has a 1-A current rating and must be fitted to a suitable heatsink.

Note that — in Figure 3's circuit — the available output current is also shunt-boosted to a total of about 5 A via Q1 and current-sensing resistor R1. At low currents, insufficient voltage is developed across R1 to turn on Q1, so all the load current is provided by the IC. At load currents of 600 mA or greater, however, sufficient voltage (600 mV) is developed across R1 to turn on Q1. Q1 thus provides most of the load currents in excess of 600 mA.

Note in the above three circuits that the ripple voltage generated across smoothing capacitor C1 is directly proportional to the regulator's output load current. As a rough rule of thumb, in a full-wave rectified power supply, operating from a 50-60 Hz power line via a step-down transformer, an output load current of 100 mA will cause a ripple waveform of about 700 mV peak-to-peak to be developed on a 1,000 µF filter capacitor.

The amount of ripple is directly proportional to the load current and inversely proportional to C1's capacitance value. Thus, the circuit in Figure 1 — which has a C1 value of

470 µF — generates a C1 peak-to-peak ripple voltage of about 1.4 V at its rated output current of 100 mA. The circuit in Figure 2 (C1 = 2,200 µF) generates a C1 peak-to-peak voltage of about 2.4 V at its rated output current of 750 mA, and the circuit in Figure 3 (C1 = 22,000 µF) generates a C1 peak-to-peak voltage of about 1.6 V at its rated output of 5 A. At the output of each circuit, these ripple values are reduced by about 80 dB by the action of IC1.

Input Signals

When using TTL, all IC input signals must — unless the IC is fitted with a Schmitt-type input — have very sharp rising and falling edges (typical rise and fall times should be less than 40 nS on LS TTL, for example). If rise or fall times are too long, they may allow the input terminal to hover in the TTL element's linear indeterminate zone (see last month's article) long enough for the element to burst into wild oscillations and generate spasmodic output signals that may disrupt associated circuitry (such as counters and registers).

If necessary, slow input signals can be converted into fast ones by feeding them to the IC's input terminal via an inverting or non-inverting Schmitt element, as shown in Figure 4. In practice, these simple circuits can be cheaply implemented by using a 74LS14 (or similar) IC that houses six Schmitt inverter elements. Two of these elements can be cascaded to make one non-inverting Schmitt element. All unused elements should be disabled by connecting their inputs directly to the 0 V rail (see the next paragraph for a deeper description).

Unused Input

Unused TTL input terminals should never be allowed to simply float, since this makes them susceptible to noise pick-up, etc. Instead, they should be tied to definite logic levels, by connecting them to V_{CC} via a 1K resistor, by shorting them directly to the ground rail, or by connecting them to a TTL input or output terminal that is already in use.

Figure 5 shows examples of the four options. The simplest option is to tie the unused input to V_{CC} via a 1K resistor, as shown in Figure 5(a). This resistor has to supply only a few microamperes of current (I_{IH}) to each input, and can thus easily drive up to 10 unwanted inputs. Alternatively, the input can be tied directly to ground, as in Figure 5(b), but in that case, an input current of several hundred microamperes (I_{IL}) may flow to the ground rail via the input.

If the unwanted input is on a multi-input gate,

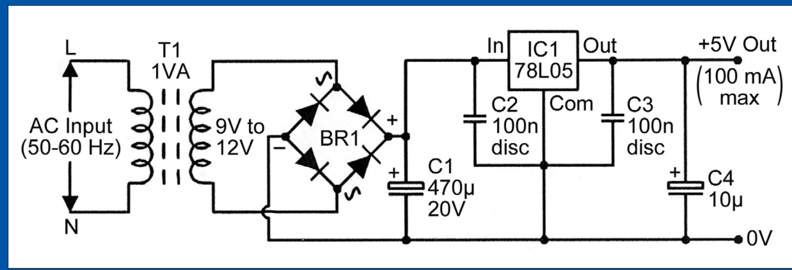


FIGURE 1. A 5 V regulated DC supply (100 mA maximum output).

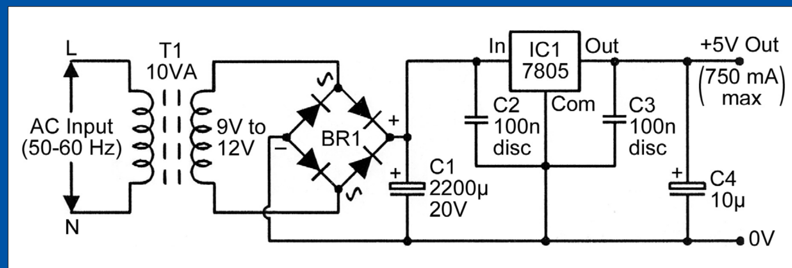


FIGURE 2. A 5 V regulated DC supply (750 mA maximum output).

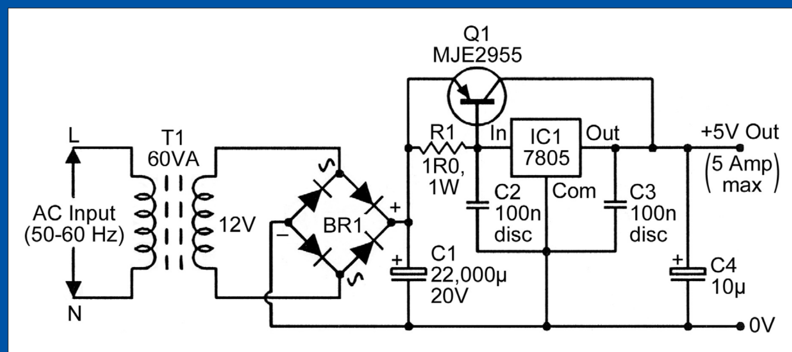


FIGURE 3. A 5 V regulated DC supply (5 A maximum output).

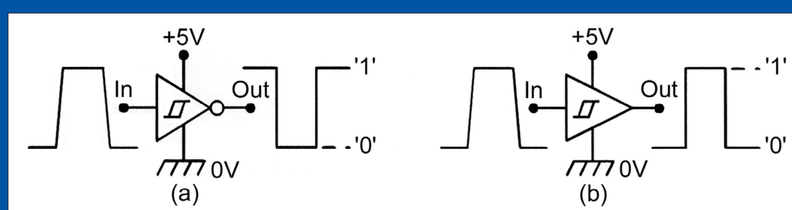


FIGURE 4. Slow input signals can be converted into fast ones via (a) an inverting or (b) non-inverting Schmitt element.

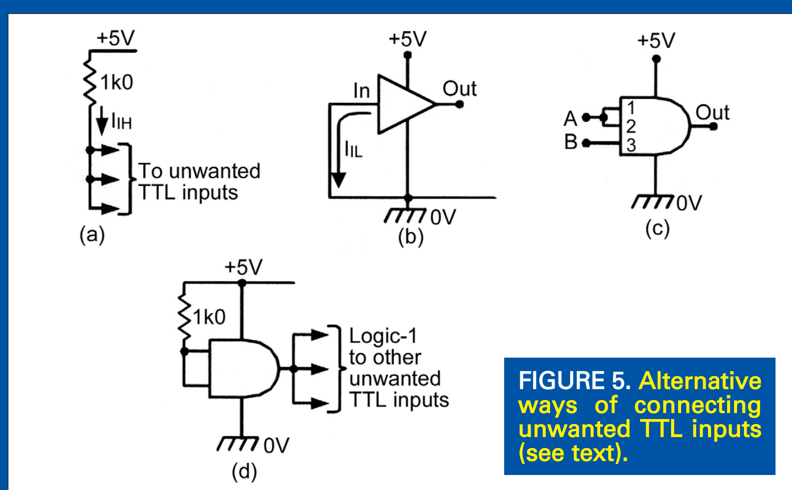


FIGURE 5. Alternative ways of connecting unwanted TTL inputs (see text).

it can be disabled by shorting it to one of the gate's used inputs, as in Figure 5(c), where a three-input AND gate is shown used as a two-input type. If the IC is a multiple-gate type, and an entire gate is unwanted, that gate should be disabled by tying its inputs high if it is a non-inverting (AND or OR) type, or shorting them to ground if it is an inverting (NAND or NOR) type. If desired, the output of this gate can then be used as a fixed logic-1 point that can be used to drive other unwanted inputs, as shown in Figure 5(d).

Interfacing

An interface circuit is one that enables one type of system to be sensibly connected to a different type of system. In a purely TTL system, in which all ICs are designed to connect directly together, interface circuitry is usually needed only at the system's initial input and final output points, to enable them to merge with the outside world via items such as switches, sensors, relays, and indicators, etc.

Occasionally, however, TTL ICs may be used in conjunction with other logic families (such as CMOS), in which case an interface may be needed between the different families. Thus, as far as TTL is concerned, there are three basic classes of interface circuit. These will now be dealt with under the headings of *Input Interfacing*, *Output Interfacing*, and *Logic-Family Interfacing*.

> **It is usually a fairly simple matter to design logic circuitry using "74-series" TTL ICs, provided that a set of TTL basic usage rules are observed."**

Input Interfacing

The digital signals arriving at the inputs of any TTL system must be clean ones with TTL-defined logic-0 and logic-1 levels and with very fast rise and fall times (less than 40 nS in LS TTL systems). It is the task of input interfacing circuitry to convert external input signals into this format. Figures 6 to 9 show a few simple examples of such circuitry.

Mechanically derived switching signals are notoriously bouncy (see Figure 1 in Part 1 of this mini-series) and must be cleaned up before being fed to a normal TTL input. Figure 6 shows a practical switch-debouncing input interfacing circuit. Here, C1 charges — with a time constant of about 10 ms — via R1-R2 when S1 is open and generates a logic-0 output via the TTL Schmitt inverter. When S1 is closed, it rapidly discharges C1 via R2, driving the Schmitt output high. The effects of any switch-generated bounce signals are eliminated by the circuit's 10 ms time constant, and a clean TTL switching waveform is thus available at the Schmitt's output.

Figure 7 shows a circuit that can be used to interface almost any clean digital signal to a normal TTL input. Here, when the input signal is below 500 mV (Q1's minimum turn-on voltage), Q1 is cut off and the inverting Schmitt TTL output is at logic-0. When the input is significantly above 600 mV, Q1 is driven on and the Schmitt output goes to logic-1. Note that the digital input signal can have any maximum voltage value, and R1 is chosen to simply limit Q1's base current to a safe value.

Figure 8 is a simple variation of the above circuit, with the transistor built into an optocoupler. The circuit action is such that the Schmitt's output is at logic-0 when the optocoupler input is zero, and at logic-1 when the input is high. Note that the optocoupler provides total electrical isolation between the input and TTL signals.

Finally, Figure 9 is another simple circuit variation, with the basic digital input signal fed to Q1's base via the R1-C1-R2-C2 low-pass filter network. This filter eliminates high-frequency components and can thus convert very dirty input signals (such as those from vehicle contact-breakers, etc.) into a clean TTL format.

Output Interfacing

Most TTL ICs have normal totem-pole output stages, but some of them have modified totem-pole outputs with three-state (tri-state) gating. A few TTL ICs have open-collector totem-pole output stages. Note that normal totem-pole outputs should not (except in a few special cases) be connected in parallel. TTL open-collector outputs *can* be connected in parallel, however, and tri-state ones can be connected in parallel under special conditions. Basic ways of using open-collector and tri-state outputs will be described in a future digital ICs article.

A normal totem-pole output stage can source or sink useful amounts of output current, and can be used in a variety of ways to interface with the outside world. A few simple examples of such circuits are shown in Figures 10 to 17. Figure 10 shows a couple ways of driving LED output indicators via non-inverting TTL elements. Note that a normal TTL output can sink fairly high load currents (typically up to 50 mA in an LS device), but has an internally limited output sourcing ability. Thus, the LED current must be limited to a safe value via R1 if it is connected as in Figure 10(a), but is internally limited in Figure 10(b).

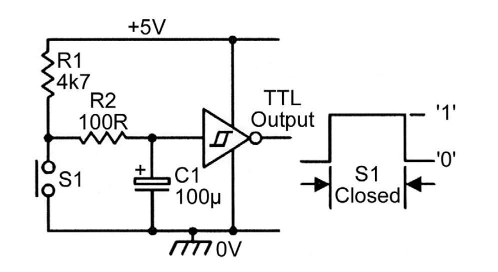


FIGURE 6. Switch-debouncing input interface.

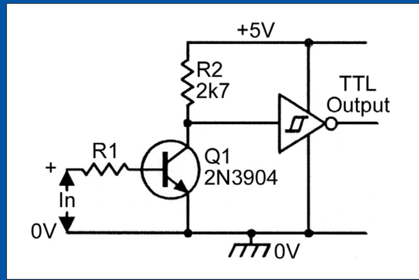


FIGURE 7. Transistor input interface.

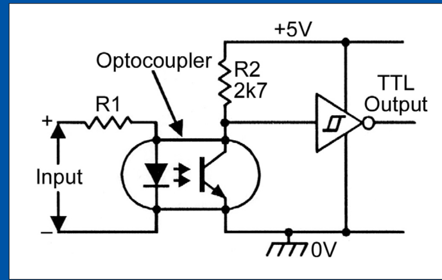


FIGURE 8. Optocoupler input interface.

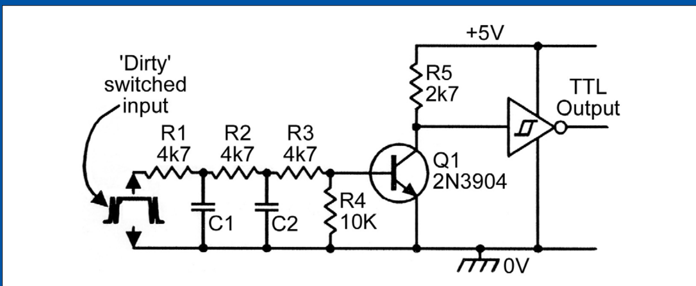


FIGURE 9. "Dirty-switching" input interface.

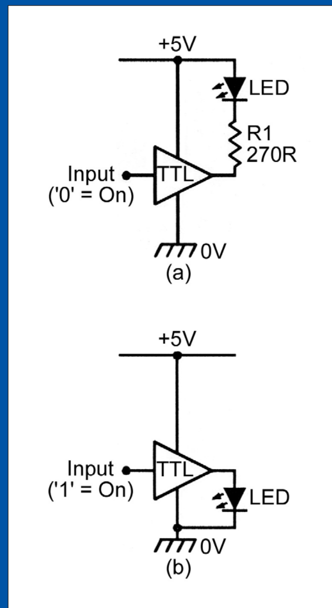


FIGURE 10. LED-driving output interfaces, using non-inverting TTL elements.

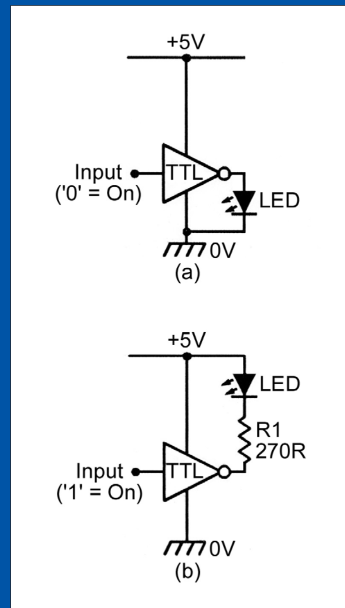


FIGURE 11. LED-driving output interfaces, using inverting TTL elements.

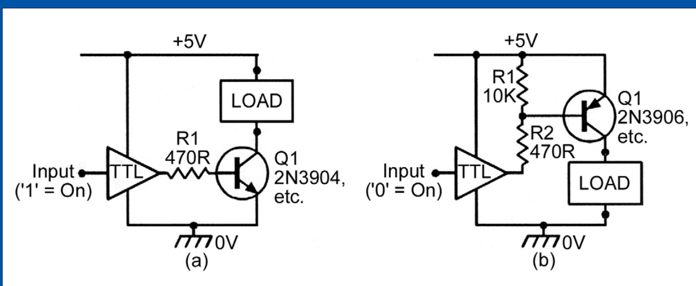


FIGURE 12. Current-boosting, load-driving output interfaces.

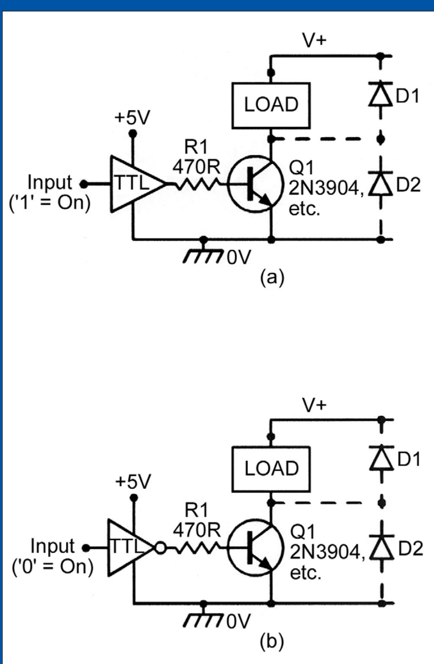


FIGURE 13. Output interface to a load with an independent positive rail.

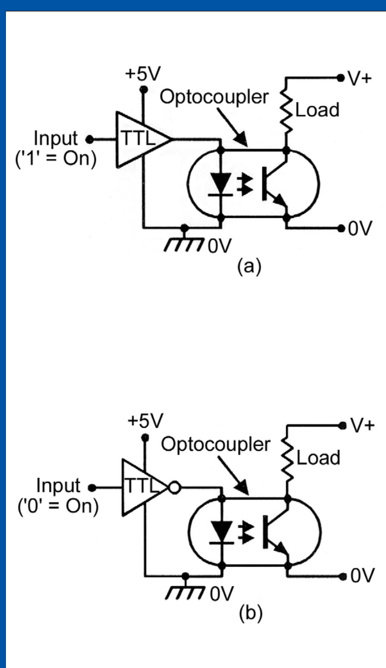


FIGURE 14. Optocoupled output interface.

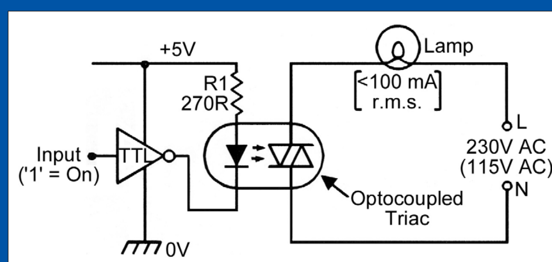


FIGURE 15. Output interface to a low-power AC lamp via an optocoupled triac.

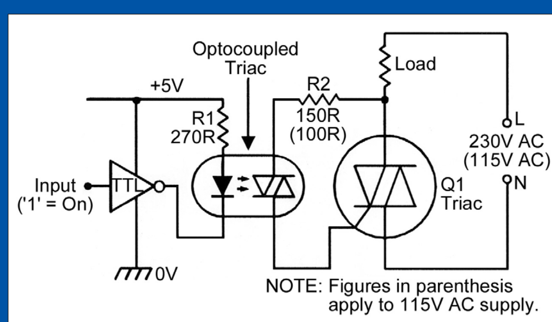
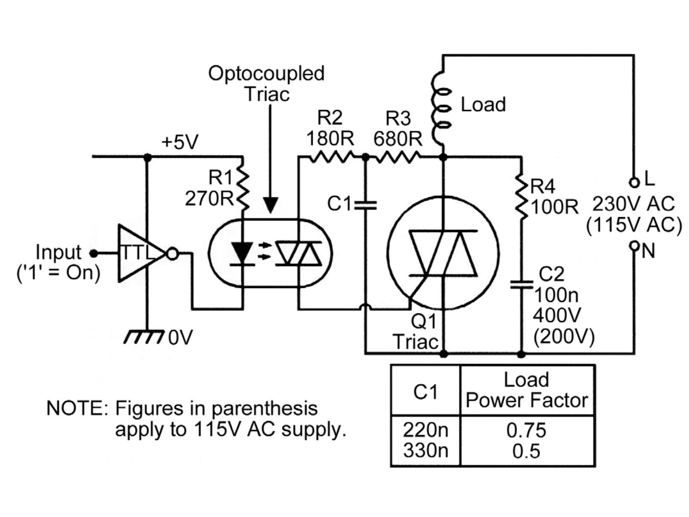


FIGURE 16. Output interface to a high-power non-inductive AC load.



C1	Load Power Factor
220n	0.75
330n	0.5

FIGURE 17. Output interface to a high-power inductive load.

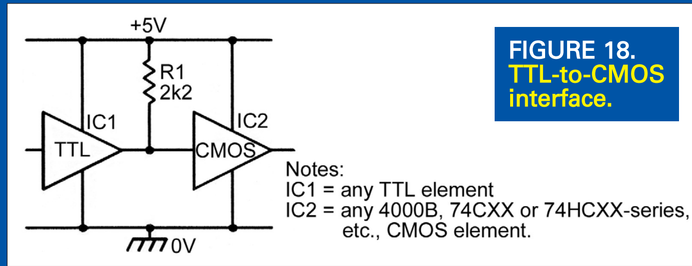


FIGURE 18. TTL-to-CMOS interface.

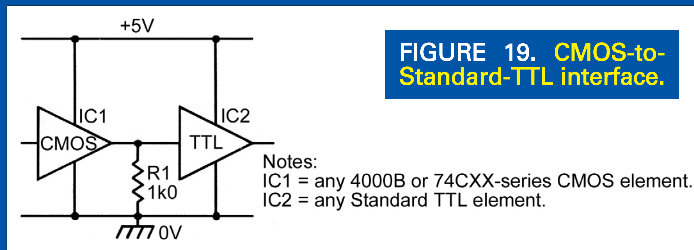


FIGURE 19. CMOS-to-Standard-TTL interface.

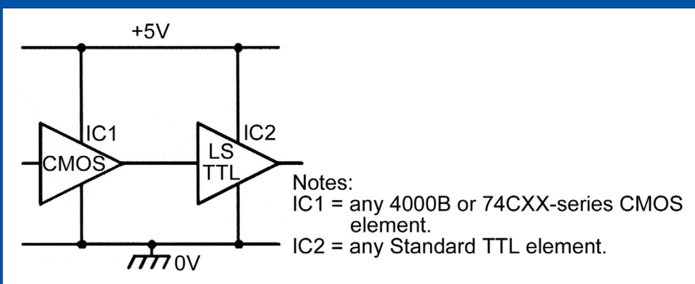


FIGURE 20. CMOS-to-LS-TTL interface.

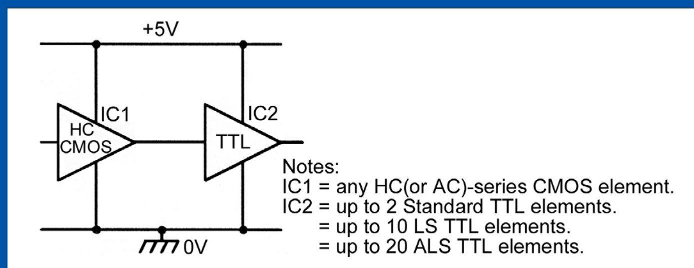


FIGURE 21. HC-CMOS-to-TTL interface.

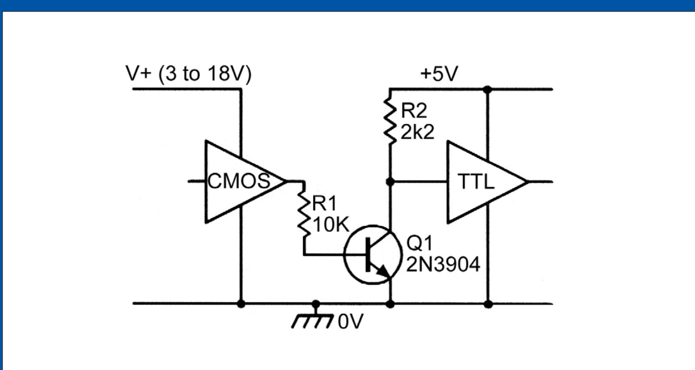


FIGURE 22. CMOS-to-TTL interface, using independent positive supply rails.

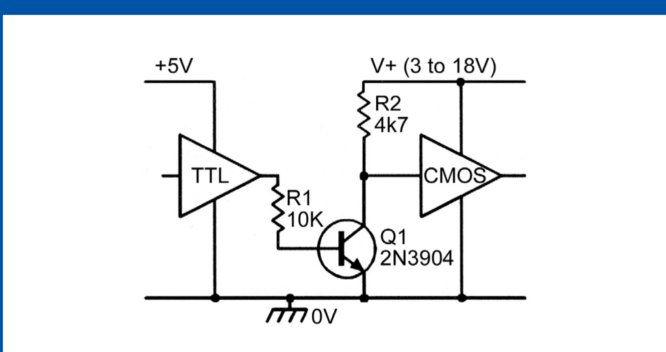


FIGURE 23. TTL-to-CMOS interface, using independent positive supply rails.

Figure 11 shows alternative ways of driving LEDs, using inverting TTL elements.

Figure 12 shows two current-boosting load-driving output interface circuits, in which the load uses the same power supply as the TTL circuit. In Figure 12(a), NPN transistor Q1 is cut off when the input of the non-inverting TTL element is at logic-0, and is driven on via R1 when the input is at logic-1. The reverse action is obtained in Figure 12(b), where PNP transistor Q1 is pulled on via R2 when the input is at logic-0, and is cut off via pull-up resistor R1 when the input is at logic-1.

Figure 13 shows two output interface circuits that can be used to drive loads that use independent positive supply rails. Q1 is turned on by a logic-1 input in Figure 13(a) and by a logic-0 input in Figure 13(b). If the external load is inductive (such as a relay or motor), the circuits should be fitted with protection diodes, as shown dotted in the diagrams.

Figure 14 shows two optocoupled output-interface circuits that can be used to drive loads that use fully independent DC power supplies. The load is turned on via a logic-1 input in Figure 14(a) and by a logic-0 input in Figure 14(b). Note that the optocoupler input (the LED) could alternatively be connected between the +5-V rail and the TTL output via a current-limiting resistor, using the same basic connections as in Figure 10(a) or Figure 11(b).

Figure 15 shows an output interface that can be used to control a low-power lamp or similar resistive load that is

driven from AC power lines and consumes no more than about 100 mA of current. This circuit uses an optocoupled triac. These typically need an LED input current of less than 15 mA and can handle triac load currents of up to about 100 mA mean (500 mA surge) at up to 400 V peak. Note that optocoupled triacs are best used to activate a high-power slave triac, which can then drive a load of any desired power rating. Figures 16 and 17 show two such circuits.

The circuit in Figure 16 is suitable for use with non-inductive loads, such as lamps and heating elements. It can be modified for use with inductive loads, such as motors, by using the connections of Figure 17. In Figure 17, R2-C1-R3 provides a degree of phase shift to the triac gate-drive network, to ensure correct triac triggering action. R4-C2 forms a snubber network, to suppress rate effects.

Logic-Family Interfacing

It is generally bad practice to mix different logic families in any system, but on those occasions where it does occur, the mix is usually made between TTL and CMOS devices that share a common 5 V power supply. In this case, the form or necessity of any interfacing circuitry depends on the direction of the interface and on the precise sub-families that are involved. Figures 18 to 21 show the four most useful types of interface arrangement.

The output of any TTL element can be used to drive any normal CMOS logic IC (including some sub-members of the 74-series) by using the connections shown in Figure 18, in which R1 is used as a TTL pull-up resistor to ensure that the CMOS consumes minimal quiescent current when the TTL output is in the logic-1 state.

Standard 4000B-series and 74CXX-series CMOS elements have very low fan-outs, and can only drive a single standard TTL or LS TTL element, as shown in Figures 19 and 20. 74HCXX-series (and 74ACXX-series) CMOS elements, on the other hand, have excellent fan-outs, and can directly drive up to two standard TTL inputs, 10 LS TTL inputs, or 20 ALS TTL inputs, as shown in Figure 21.

In cases where the TTL and CMOS ICs use individual positive supply rails (5 V for TTL, 3-18 V for CMOS), an interface can be made between the two systems by using a direct-coupled NPN transistor as a level shifter between them, as shown in Figures 22 and 23. (These simple circuits may need some refining if they are to be used at frequencies above a few hundred kHz.)

Finally, note that if the TTL element has an open-collector, totem-pole output, a direct interface can sometimes be made between the TTL output and the input of an individually-powered CMOS element, etc. The basics of this technique will be described in next month's installment of this four-part mini-series. **NV**

ELECTRONET

>>>> go to www.nutsvolts.com — click Electro-Net <<<<

Elf Turns 30

■ by Robert Armstrong ------------------------------> **Part 2**

Welcome back! Last month, we printed the schematics for the Elf 2000 and discussed the operation of every part of the circuit. This month, we'll print the parts list, talk about the construction and testing of your Elf 2000, say a few words about the available software, and then finish up with ideas for expansion.

Construction

You can construct the Elf 2000 using wire wrap or point-to-point wiring, just like the original Elf, but it's a lot more fun if you use a PCB (printed circuit board). A pre-made PCB is available from **http://elf2k.SpareTimeGizmos.com** and, not only does it make wiring faster and easier, but it makes it a lot less error prone, as well.

Assembling the Elf 2000 is a straightforward process of inserting the parts into the PCB and soldering them. Here are a few tips that will help you avoid any problems:

• Twenty-one 0.1 µF 50 VDC monolithic bypass capacitors are used in the Elf 2000. On the PCB, these are identified only by a box on silkscreen.

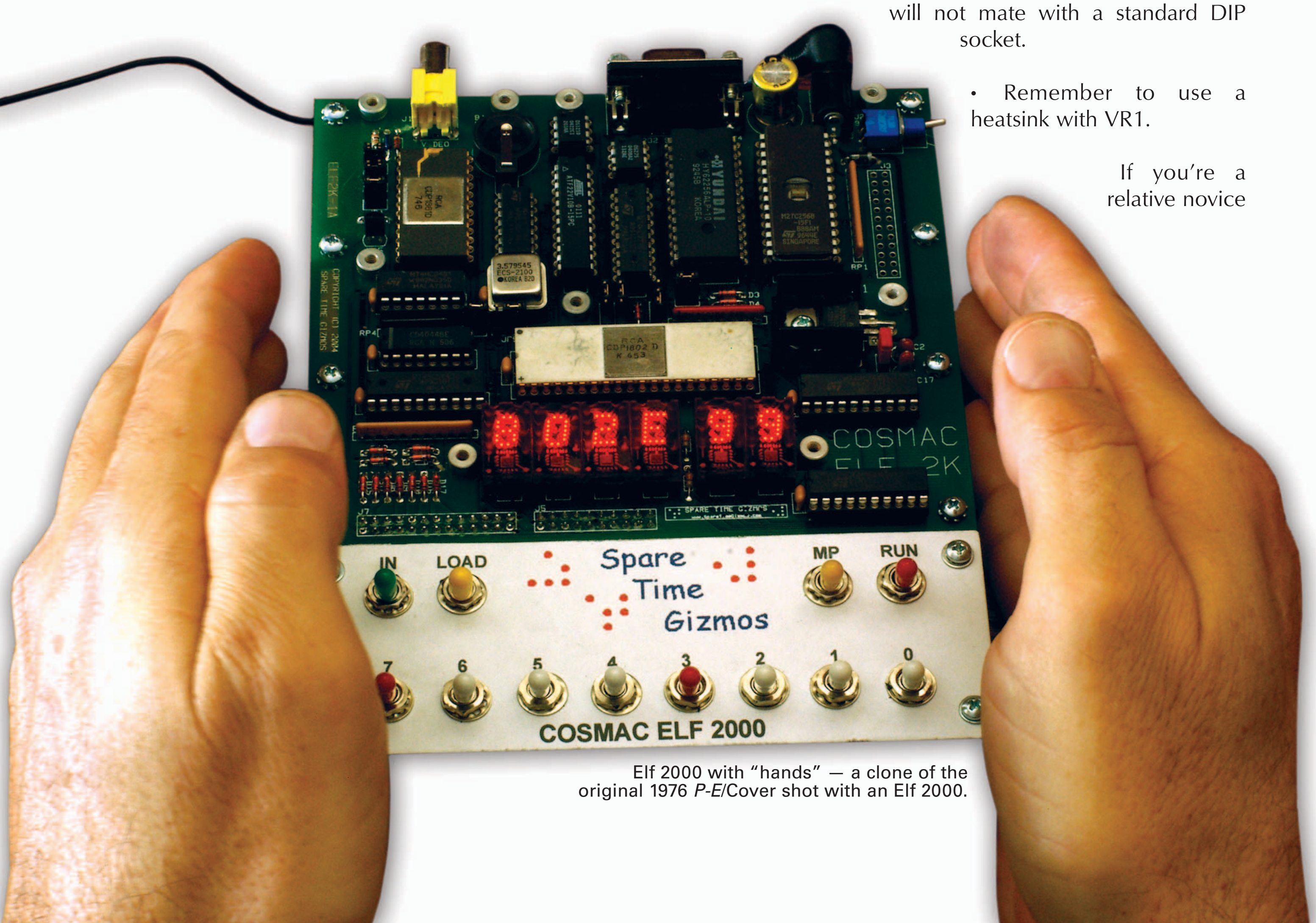
• Notice that capacitors C1 and C2 are polarized devices and must be installed correctly. The polarization is shown on the silkscreen of the PCB.

• Header J5 (the switch panel connector) should be mounted on the bottom (solder side) of the PCB with the pins facing "down." Because of the limited space available, don't use a shrouded header for J5.

• If you intend to use the STG1861 "Pixie" emulator, then do not install a DIP socket at U2 (the CDP1861 socket). The STG1861 daughter board uses special 0.1" female headers and will not mate with a standard DIP socket.

• Remember to use a heatsink with VR1.

If you're a relative novice



Elf 2000 with "hands" — a clone of the original 1976 *P-E*/Cover shot with an Elf 2000.

Jumper	Default Setting
JP1	Installed if you have a CDP1861/STG1861.
JP2	Installed (connects *INPUT* to EF4).
JP3	Unused (doesn't exist on the current PCB).
JP4	Installed (automatically start the monitor after power up).
JP5	Unused for CDP1802 CPU.
JP6	Connect RxD to EF3.
JP7	Same as JP1.
JP8	Same as JP1.
JP9	Invert RxD.
JP10	Invert RxD.

TABLE 1. Default Jumper Settings.

at construction, you might want to download the *Elf 2000 User's Manual* from **http://elf2k.SpareTimeGizmos.com** and read the Assembly section. It contains many tips for the beginner on parts selection, preparation, soldering, testing, and debugging.

Although the complete Elf 2000 may seem complex, it breaks down easily into many separate subsystems. Most of the subsystems in the Elf 2000 — including the video, EPROM, battery backup, switches, address and data displays, and RS232 port — are optional and can be omitted to simplify construction. In a few cases, such as the battery backup, it's necessary to add a few jumpers to the Elf 2000 to ensure that the rest of the circuit still works when a particular part is omitted. *The Elf 2000 User's Manual* gives the details on this and also contains a list of the exact parts that can be omitted for each subsystem.

Check Out

After you finish assembly, apply power before you install any ICs and use a DC voltmeter to verify that the voltage between pins 20 (negative) and 40 (positive) on the microprocessor (CDP1802) socket is between 4.9 and 5.1 volts. Remove power and install all ICs except the CDP1802, CDP1861 (if you have one), and the TIL311 displays. Turn on the power and check that the current consumption is about 100 mA. If you don't get these results, then stop and figure out what's wrong before proceeding. Finally, turn off the power once more and install the remaining parts.

If your Elf 2000 has a switch panel, then you can do a few simple tests with the switches to verify that everything is working correctly. If you don't have a switch panel, then just skip ahead to the paragraph on the POST in the next section.

Ensure that all switches are set to the OFF position, including D0-D7, and turn on the power — only the SC0 LED should be illuminated and the address display should show 0000 (the data display may be any random value). Flip LOAD to the ON position and the LOAD LED should light. Press or flip the INPUT switch and the display should read 0000 00.

Now set all the D0-D7 switches to ON and press INPUT again; the display should now read 0001 FF. Next try $A5 on the data switches and press INPUT and the display should show 0002 A5. Finally, try $5A and INPUT and the display should show 0003 5A.

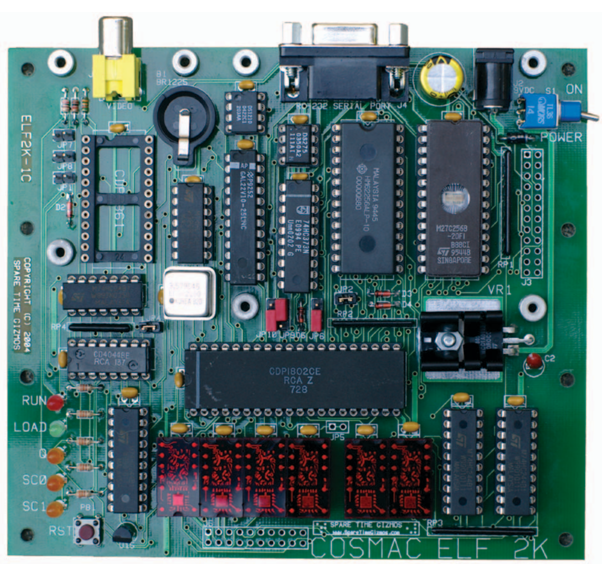


FIGURE 1. A closeup of the PC board alone.

Set LOAD to OFF, Memory Protect to ON, then flip LOAD back to ON and the display will show 0000 5A. Press INPUT and the display will show 0000 00 — this is the contents of RAM location 0. Try pressing INPUT three more times and you'll see the next three bytes that you just entered; 0001 FF, 0002 A5, and 0003 5A. Congratulations — your Elf 2000 can store programs in memory!

Software

POST

If you're using the Elf 2000 EPROM, then first ensure that all the PCB jumpers are set as shown in Table 1. These settings are necessary for the monitor to work and you can find more information about them, including photos of the correct settings, in the *Elf 2000 User's Manual*.

Connect an RS232 terminal or your PC to the Elf 2000 serial port and set the terminal or terminal emulator software for 2400 baud, 8-N-1 (eight data bits, no parity and one stop bit). If you have a switch panel, set all Elf 2000 switches to OFF except RUN, which must be ON, and turn on the power. The RUN LED should light and the data display should show 99, followed by 98, and steadily count down to 16.

This count is the "Power On Self Test" executing, and if the count stops at any point before 16, then you have a hardware problem. If you need it, the *Elf 2000 User's Manual* contains a full list of the 30 some POST codes to help you debug your Elf 2000. POST code 16 means that the monitor is ready for auto baud; press the ENTER (carriage return) key on your PC or terminal and you should see something like this:

```
COSMAC ELF 2000 EPROM V85 CHECKSUM
7D14 SRAM 32K INITIALIZED
Copyright (C) 2004-6 by Spare Time
Gizmos. All rights reserved.
ElfOS BIOS Copyright (C) 2004 by
Mike Riley.
For help type HELP.
>>>
```

Congratulations! If your Elf 2000 passes the Power On Self Test, you

can be pretty certain that your hardware is okay.

Monitor

In addition to the Power On Self Test just described, the Elf 2000 EPROM contains a simple monitor program. This monitor allows you to examine and deposit in memory, manually read and write from I/O ports, start programs loaded in RAM, set breakpoints in RAM programs, examine registers, and continue execution after a breakpoint.

The monitor also contains a HEX file loader that allows you to download Intel format HEX files over the serial port directly from the PC. The monitor allows you to execute a more extensive test of the memory and video systems than the POST. And finally, the monitor contains a bootstrap and BIOS for the Elf disk operating system that's mentioned below.

It's worth noting that the complete source code for all of the Elf 2000 EPROM contents, including the monitor, BIOS, and languages, is available to the experimenter from **http://elf2k.SpareTimeGizmos.com**. You're free to make changes, customize it as you like, and rebuild your own EPROM.

Languages

Besides the POST and monitor

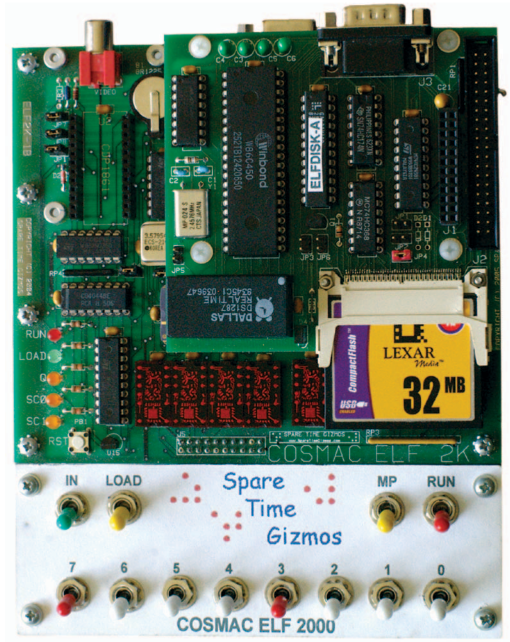


FIGURE 2. A view showing the disk board installed.

Parts List

LOC	Supplier	Part No.	Description
<u>Resistors</u> *(Unless noted, all resistors are 1/8W 5% carbon composition.)*			
R1, R5-R8			330 ohm
R2			1K
R3			200 ohm
R4			3.3K
RP1, RP3			10 pin 9x10K SIP Resistor
RP2			8 pin 7x10K SIP Resistor
RP4			6 pin 5x10K SIP Resistor
<u>Capacitors</u>			
C1			1,000 µF 16V Aluminum Capacitor
C2			10 µF 6V Tantalum Capacitor
C3			*Unused*
C4-C24			0.1 µF Mono Ceramic Bypass Capacitor
<u>Integrated Circuits</u>			
U1	Intersil	CDP1802ACD3	Eight bit Microprocessor (see text)
U2	RCA	CDP1861D	Video Co-processor (see text)
U3		27C256	32Kx8 CMOS EPROM
U4		62256LP	32Kx8 Low Power SRAM
U5	Dallas	DS1210	Nonvolatile SRAM Controller
U6		74HC373	Octal D Latch
U7	Atmel	ATF22V10CQZ	Flash GAL ("Quarter Power")
U8		74HC04	Hex Inverter
U9		CD4044	Quad S-R Latches
U10		74HC74	Dual D Flip Flop
U11			*unused*
U12-U14		74HC244	Dual Quad Buffer
U15	Dallas	DS1233	5V 10% TO-92 EconoReset
U16	Dallas	DS275	RS-232 Transceiver

*Most integrated circuits — except the 1802 and 1861 — are available from Mouser (**www.mouser.com**), Digi-Key (**www.digikey.com**), and/or Jameco (**www.jameco.com**). The Dallas parts can be ordered directly from the Dallas/Maxim web page at **www.maxim-ic.com***

LOC	Supplier	Part No.	Description
<u>Semiconductors</u>			
D1			*Unused*
D2-D4, D6		1N914	Small Signal Switching Diode
D5		1N4001	50PIV 1A Power Diode

Parts List continued ...

LOC	Supplier	Part No.	Description
VR1		7805	5.0V Fixed Output TO-220 Regulator
DISP1-6		TIL311	Hexadecimal LED Display
LED1			T1 LED (Red)
LED2			T1 LED (Green)
LED3-LED5			T1 LED (Orange)
OSC1	ECS	2100A-035	3.579545 MHz DIP8 Crystal Oscillator (Digi-Key P/N XC235-ND)
Miscellaneous			
B1		CR1225	Lithium Coin Cell
		BH1225	Coin Cell Holder (Mouser P/N 500K)
PB1	Panasonic	EVQ-PAD05R	Push Button Switch (Digi-Key P/N P10890S-ND)
S1	C&K	T101MH9ABE	SPDT Toggle Switch (Digi-Key P/N CKN1067)
		ELF2K-1B	PCB (elf2k.SpareTimeGizmos.com)
J1			PCB Mount Phono Jack
J2			Coaxial Power Jack (Digi-Key P/N CP-102AH)
J3			24 pin Dual Row Stacking Header (elf2k.SpareTimeGizmos.com)
J4			PCB Mount DE9F
J5			20 Pin 0.1″ Dual Row Header
JP1, JP2, JP4, JP5, JP7, JP8			2 Pin 0.1″ Header and Shorting Plugs
JP6, JP10			3 Pin 0.1″ Header and Shorting Plugs
	Keystone	1560A	#4-40 Swages, 0.125″ x 0.250″ (Mouser P/N 534-1560A)
			Heatsink for VR1 (Digi-Key HS104-1)
			#4-40 Mounting Hardware for VR1
			Machined Pin DIP Sockets for ICs
Switch Panel (Optional)			
D0-7, RUN, MP	Jameco	76523CR	SPST Toggle Switch
LOAD	Jameco	21936CR	SPDT Toggle Switch
INPUT	Jameco	28062CR	SPDT Push Button Switch
J5			20 Pin 0.1″ Dual Row Header
			6″ 20 Conductor Ribbon Cable w/ IDC Connectors
			5 1/2″ x 2″ x 1/8″ ABS Plastic for the Switch Panel
			6 3/4″ x 1″ x 3/8″ Acrylic Plastic for the Side Rails
			Front Panel Decal (from the Elf 2000 Manual)

program, the EPROM also contains three Elf languages: Basic, Forth, and an Editor/Assembler, all written by Mike Riley. A discussion of these languages would take up way more room than we have for this article, and you should read the *Elf 2000 User's Manual* to learn more about them.

ElfOS

Finally, although it isn't part of the EPROM, it's worth mentioning that Mike Riley has also written an entire disk operating system for the 1802 that runs on the Elf 2000 and other similar hardware. To run it you'll need the Elf 2000 Disk, UART and RTC expansion card described below, and a mass storage device such as a Compact Flash card. Check out Mike's web page (**www.elf-emulation.com**) for all the details.

Expansion

If you were looking closely, you may have noticed that the Elf 2000 has a 24 pin expansion connector on the right side. This connector is designed to interface to approximately 3" square daughter cards that stack up on top of the Elf 2000 — some of the expansion cards that are currently available include:

• *A Disk, UART, and RTC Card* — This option contains an IDE disk interface, Compact Flash socket, a real UART (to eliminate the software overhead of a bit banged serial port), and a real time clock. This board is needed to run ElfOS.

• *A General-purpose I/O Card* — This card contains an 8255 PPI chip that implements three eight-bit general-purpose I/O ports, a PS/2 keyboard interface that converts the PS/2 keyboard to parallel ASCII for the Elf, and a speaker that can play various tones and simple music.

• *A Video Card* — This card implements an 80 column by 24 line text video display and can be used together with the PS/2 keyboard interface on the GPIO card to replace the Elf 2000 console terminal. The standard Elf 2000 EPROM contains the necessary firmware to emulate a VT52 type terminal using this hardware.

Now What?

Okay, you've made it this far; now what? If you do a Google search for "COSMAC Elf" you'll find literally thousands of hits, including copies of 1970's magazine articles, projects, and software. All of this should work on your Elf 2000. You can also join the Yahoo! COSMAC Elf group (**http://groups.yahoo.com/group/cosmacelf/**) and browse their Files and Messages archives. And, in addition, there's the Spare Time Gizmos group (**http://groups.yahoo.com/group/sparetimegizmos/**) specifically for Elf 2000 and projects.

If you haven't done it already, be sure to download the 100+ page *Elf 2000 User's Manual* from **http://elf2k.SpareTimeGizmos.com**. You can also find the original schematics, source code, and HEX files for the EPROM, and PLD programming files there. And finally, you can purchase bare PCBs, partial kits including 1802 chips and TIL311 displays, pre-programmed GALs and EPROMs, and other Elf 2000 parts from Spare Time Gizmos.

Mike Riley's fine software — including the Basic and Forth interpreters, interactive Editor/Assembler, and the entire ElfOS Disk Operating System — are available from his website at **www.elf-emulation.com**.

If you don't already have a terminal emulator program for your PC, then two excellent choices are Hyperterm and Kermit (**www.columbia.edu/kermit/**). **NV**

■ FIGURE 1. Completed boost power supply.

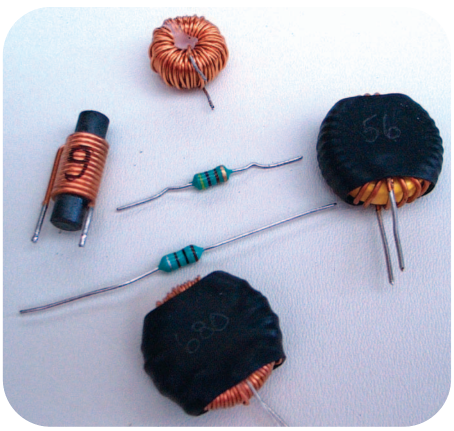
The High Voltage PIC
PART 1

— by Robert Lang

CAN YOU GENERATE 200 VOLTS DC FROM A FIVE-VOLT DC PIC POWER SUPPLY? Yes, you can, and I will tell you how. In this two-part series, we will build a 200-volt DC boost power supply driven by an 18F2455 Microchip PIC microprocessor. The completed power supply (shown in Figure 1) will generate between five and 200 volts DC from a five-volt, DC input.

This device can be used for any application where high voltage and low current is needed, such as PIC programming (13 volts), neon indicator bulbs (120 volts), or nixie tube displays (170 volts).

■ FIGURE 2. An inductor assortment.



In this first part, we'll cover the theory of how a boost power supply works. You'll learn how to use pulse width modulation (PWM) to get the 18F2455 PIC to output a square wave with a given period and duty factor, and we'll look at how the PWM's "ON" time affects voltage. We will cover the important parameters in designing a boost power supply and use the free LTSPICE program to come up with a possible design.

Next month, we will implement this design on a printed circuit board. We will use a liquid crystal display (LCD) to show important power-supply parameters. Using a free C compiler, we'll write software to drive A/D conversion, control PWM, and output information to the LCD.

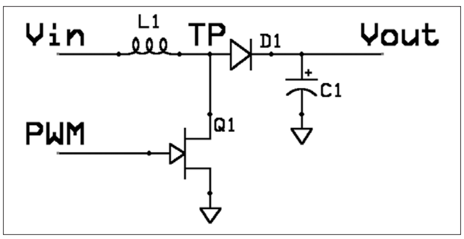
Boost Power Supply Theory

A boost power supply is a type of switching power supply that generates a higher output voltage than the supply voltage. In addition to stepping up the voltage, it can also change the sign of the voltage input. These tricks are very useful for generating a local source of other voltages from a standard +5 VDC supply.

A boost converter can have as few as four basic components: a semiconductor switch, a diode, an inductor, and a capacitor. The semiconductor switch may be a unipolar device, such as a MOSFET, or a bipolar transistor. The benefit of using a unipolar device is the absence of stored carriers and, therefore, theoretically instantaneous switching transients that are limited only by small parasitic capacitances.

If you are like me, you may not have a large assortment of inductors in your parts box and you may not use them a lot in your circuits. Let's take a more detailed look at inductors. Inductance (measured in Henries) is an effect which results from the magnetic field that forms around a current carrying conductor. Inductance can be increased by looping the conductor into a coil which causes magnetic flux from adjacent loops of the conductor to link.

■ FIGURE 3. Basic boost power supply schematic.



An inductor is usually constructed as a coil of conducting material, typically copper wire, wrapped around a core of ferrous material, which is highly permeable to magnetic flux. The coils and the core's permeability amplify the current's magnetic field, increasing inductance. Some small inductors look similar to resistors and use a similar color scheme. Other inductors look like wire-wrapped donuts (toroids), as shown in Figure 2.

Two (or more) inductors that have coupled magnetic flux form a transformer — a fundamental component of every electric power grid. An inductor is used as the energy-storage device in the boost power supply. The inductor is energized for a specific fraction of the regulator's switching frequency — the "ON" time — and de-energized for the remainder of the cycle. When we select the inductor to use in our boost power supply, we want to make sure the inductor can handle the peak current. The peak current in the inductor of the boost power supply can be written as a function of the input voltage, semiconductor ON time, and inductance:

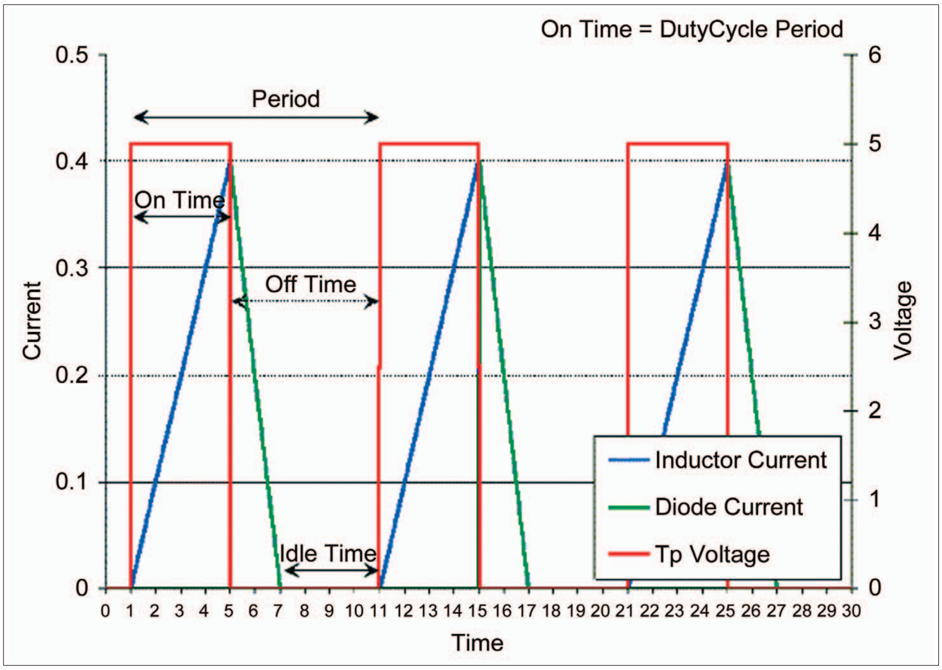
Max_Current = Input_Voltage* ON_Time/Inductance
[Equation 1]

The basic circuit is shown in Figure 3 and operates as follows:

1. The inductor current ramps up while the MOSFET is conducting (TP voltage near zero). When the MOSFET is conducting, the source voltage is applied across the inductor, and the rate of rise of inductor current depends on the source voltage Vin and inductance L1. If the source voltage remains constant, the rate of rise of inductor current is positive and remains fixed, so long as the inductor has not saturated.

2. When the MOSFET is off, the voltage at TP rises rapidly as the inductor tries to maintain a constant current. The diode turns on, and the inductor dumps current into the capacitor, increasing the output voltage to more than the input voltage.

3. The switching of the MOSFET will be controlled by a microprocessor using PWM. PWM will control the duty cycle, D, which is the fraction of the time the output is high. The value of D varies such that 0 < D < 1. The output voltage is lowest when D = 0. At zero D, the output voltage equals the source voltage. As D approaches unity, the output voltage tends to infinity. Usually D is varied such that 0.1 < D < 0.9.



■ FIGURE 4. Showing boost power supply waveforms.

A boost power supply can be driven in discontinuous conduction mode (DCM) or continuous conduction mode (CCM), depending on whether the current in the inductor is allowed to go to zero or not. In DCM, the semiconductor is switched on only after the inductor current has fallen to zero. In CCM, the switch is turned on before the current through the diode reaches zero. We will operate in DCM because we want to have the largest range of duty cycles. Also, the equations for DCM are somewhat simpler. The basic waveforms for DCM are shown in Figure 4.

Reference 1 gives the following equation for the boost power supply output voltage, in DCM, as a function of the input voltage, load resistance, semiconductor ON time, period, and inductance:

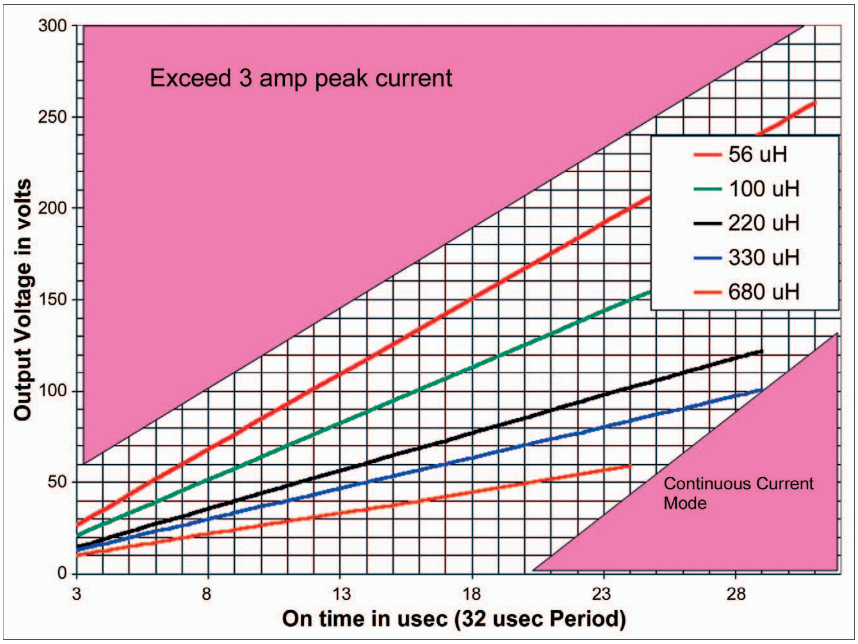
Output_Voltage = Input_Voltage* (1 + SQRT(1+(4*On_Time/Period))/ K)/2, [Equation 2]

where

K = 2*Inductance/(Output_ Load_Resistance * Period) [Equation 3]

A spreadsheet to calculate the output voltage, peak current, current fall time, and other important boost power supply parameters is available from Reference 2. I generated Figure 5 using the spreadsheet, but it assumes a

■ FIGURE 5. Output voltage versus ON time and inductance for a 32 μs period.

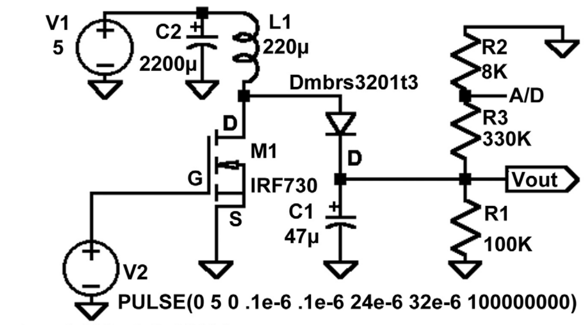


■ TABLE 1. The Boost Calculator Spreadsheet.

BOOST CALCULATOR		
INPUT		
INDUCTOR	220.00	μH
ON TIME	24.00	μsec
PERIOD	32.00	μsec
VIN	5.00	volts
INDUCTOR EFF	70.00	%
R3	330000.00	ohms
R4+R10	8000.00	ohms
RLOAD	34000.00	ohms [1]
DESIRED RIPPLE	0.02	volts
OUTPUT		
POWER OUTPUT	0.95	watts
SWITCHING FREQ	31250	Hz
PEAK CURRENT	0.55	amps
VOUT	180.27	volts
FALL TIME	0.68	μsec
RLOAD total	30892.47	ohms
A/D AT VOUT	873	counts
DC LOAD CURRENT	0.005835	amps
FILTER CAPACITOR	25.16	μF
[1] Equivalent to two nixie tubes.		

MICROPROCESSOR FREQ	48 MHz Tosc = 0.0208 μs						
CCPR1L:CCP1CON(4:5)	512	512	512	256	128	64	72
TIMER PRESCALER	16	4	1	1	1	1	16
PR2 VALUE	128	128	128	128	128	128	23
PWM FREQUENCY (Hz)	5814	23256	93023	93023	93023	93023	31250
PWM PERIOD (μs)	172.0	43.0	10.8	10.8	10.8	10.8	32.0
PWM ON TIME (μs)	170.7	42.7	10.7	5.3	2.7	1.3	24.0

■ TABLE 2. Pulse width modulation parameter table.




```
.tran 0 100e-3 0 .00001
.INCLUDE C:\Program Files\LTC\SwCADIII\lib\IRF730.SPI
.MODEL Dmbrs3201t3 d
+IS=2.18777e-06 RS=0.01 N=1.96375 EG=1.10324
+XTI=1.59554 BV=200 IBV=1e-05 CJO=1.6989e-10
+VJ=0.4 M=0.356544 FC=0.5 TT=4.54449e-08
+KF=0 AF=1
```

■ FIGURE 6. LTSPICE model of boost power supply.

perfect inductor. One can use the spreadsheet to see the effects of inductor value, ON time, voltage ripple, and load on the peak current, output voltage, and output capacitor.

Also shown in Figure 5 are two limiting conditions. First, if the ON time and inductor combination are too large, the power supply will cease to operate in DCM and will start operating in CCM. Second, if the inductor value is too small, the peak current will exceed the limiting current of one of the components.

A boost converter can have as few as four basic components: a semiconductor switch, a diode, an inductor, and a capacitor.

An Introduction to Pulse Width Modulation

Most Microchip microprocessors have a PWM feature. PWM is a process by which the ON portion of a square wave is varied as shown in Figure 4. For the PIC182455 microprocessor, the PWM output is on Pin 13 (RC2). PWM can be described by the two parameters: period and duty cycle. The period and ON time are controlled by loading certain values into registers on the microprocessor. The period is set by using the following equation:

PWM Period = (PR2+1) * 4 * Tosc *TMR2
[Equation 4]

Here, PR2 is the TIMER2 period register, and TMR2 is the TIMER2 prescaler register. Tosc is the period of the system clock, that is, (clock rate)$^{-1}$. For example, for a 48 MHz clock rate, Tosc would be 1/48 (= 0.0208) microseconds.

The ON time is calculated from the following equation:

PWM ON Time = (CCPR1L:CCP1CON<5:4>) * Tosc *TMR2
[Equation 5]

where CCPR1L:CCP1CON<5:4> is a 10-bit number consisting of the contents of the CPR1L register and bits 5 and 4 of the CCP1CON register.

If the ON time is longer than the PWM period, the CCP2 output pin will never be cleared. Table 2 shows the behavior of the PWM period and ON time for various register values. This table shows that increasing the product of PR2 and TMR2 increases the PWM period. Increasing the contents of CCPR1L and CCP1CON increases the ON time.

One common application of PWM signals is motor control. By varying the ON time of a PWM signal sent to a motor, you can vary the effective power of the signal and thereby slow the motor down or speed it up. We will use the PWM signal to control the voltage output of the boost power supply.

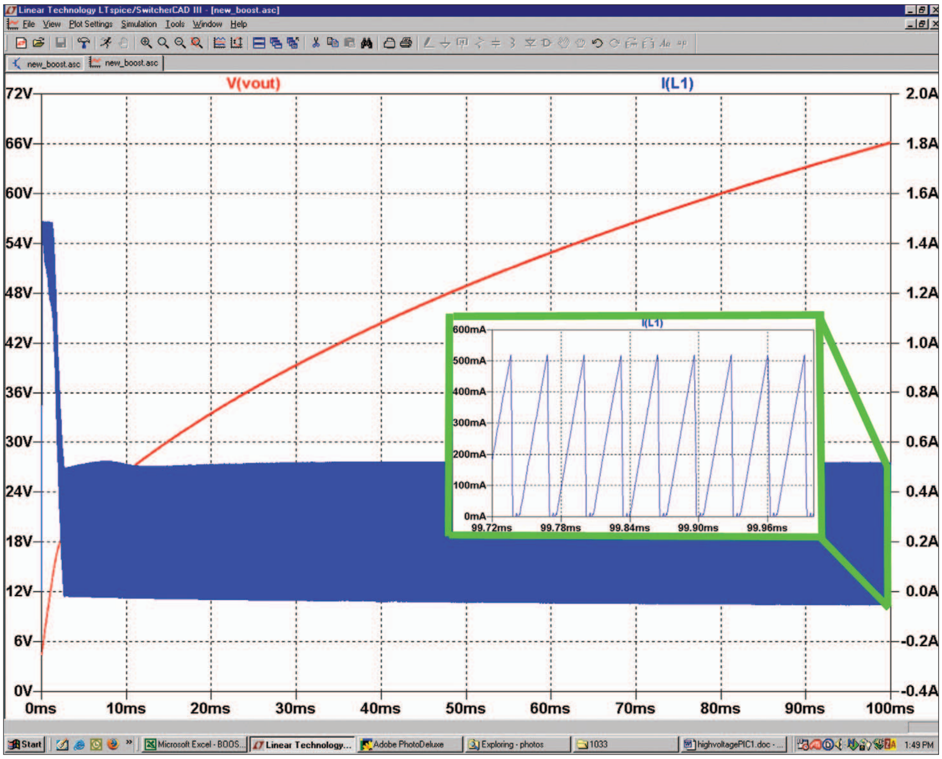
Circuit Design

In order to build the circuit, there are a number of interacting variables that need to be optimized to determine the final output voltage and the response to the requested voltage. Some of these variables are:

1. Value of L1 inductor (inductor rated at peak current)
2. Value of C1 capacitor (capacitor rated at peak voltage)
3. PWM period
4. PWM ON time
5. Values of R3 and R4 to limit A/D value to 0-5 volts

Because of the number of interacting values, I decided to use the computer program SPICE to analyze the circuit before building it on a breadboard. SPICE stands for ***S***imulation ***P***rogram with ***I***ntegrated ***C***ircuit ***E***mphasis. SPICE is a general-purpose circuit simulation program developed at the University of California, Berkley for nonlinear DC, nonlinear transient, and linear AC analyses. Circuits may contain resistors, capacitors, inductors, independent voltage and current sources, and the five most common semiconductor devices: diodes, BJTs, JFETs, MESFETs, and MOSFETs.

I had not used SPICE before, but there was an excellent article in the July '02 issue of *Nuts & Volts* [Reference 3]. The LTSPICE program mentioned in the article is available as a free download at Reference 4. The LTSPICE program now includes the SWITCHERCAD III graphical user interface that allows one to draw a schematic and analyze it with LTSPICE without ever having to enter a



■ FIGURE 7. Output of LTSPICE model.

SPICE command. Figure 6 shows the LTSPICE model that I arrived at after some trial and error.

The LTSPICE model shows two voltage sources. One is the five-volt DC supply. The second voltage source is described by the PULSE(0 5 0 1E-6 1E-6 24E-6 32E-6 1000000) command. The PULSE statement describes a square wave pulse that has an ON time of 24 microseconds and a period of 32 microseconds. The ".TRAN" command tells SPICE to run a 100 millisecond transient using 10 ms steps.

R1 is a load resistor standing in for the power supply load. R2 and R3 form a resistor voltage divider network that scales the output voltage to 0-5 volts.

There was no SPICE model included for the IRF730 MOSFET with the LTSPICE program package. The device vendor does have a SPICE model for the IRF730, called IRF730.SPI, which can be downloaded from Reference 5.

There are two types of third-party SPICE models: those described with a .MODEL statement and those defined with a .SUBCKT statement. Models given as .MODEL statements are for intrinsic SPICE devices like diodes and transistors. The .MODEL statement gives the parameters for the specific component. The behavior of the device is already known by SPICE and only the parameters need to be given to finish specifying the component's electrical characteristics. Figure 6 shows the .MODEL statement for the MBRS3201T3 Schottky diode [Reference 6].

In contrast, models given by .SUB CKT statements define the modeled component by a collection of circuitry of intrinsic SPICE devices. For example, the SPICE model of an op-amp would be given as a sub-circuit. The IRF730 is also described by a SUBCKT statement.

Adding an IRF730 MOSFET that's described by SUBCKT statements to the circuit is a little tricky. It's done as follows:

1. Add an instance of symbol NMOS to your schematic.

2. Move the cursor over the body of the newly placed N-channel MOSFET symbol instance and press <Ctrl>RightMouseButton. A dialog box will appear. Change Prefix: MN to Prefix: X. This causes this instance of the symbol to netlist as a sub-circuit instead of as a MOSFET.

3. Edit the value "NMOS" to "IRF730" to coincide with the name given on the .SUBCKT line.

4. Add the SPICE directive ".INCLUDE C:\Program Files\LTC\SwCADIII\lib\IRF730.SPI" to the schematic to point to the file containing the SUBCKT information.

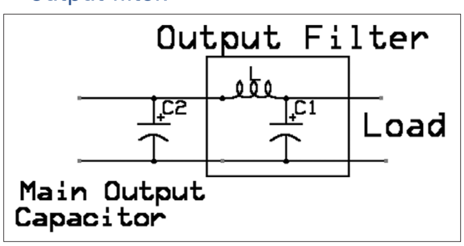
Hit the SIMULATION/RUN menu item, and the output should look like Figure 7. Figure 7 shows that the output voltage increases to approximately 66 volts over the first 100 milliseconds after power is turned on. If you examine I(L1), you will see that when the power is first turned on, there is a large transient in the current through the inductor. The large transient occurs if the ON time of the semiconductor switch is constant at 24 µs. We can reduce the required peak current capacity of the inductor by using a software "soft start." In the "soft start" algorithm, ON time is initially set to zero, then increased slowly until the desired voltage is reached.

If you examine the blowup of the last part of the startup, you will see that the current through the inductor returns to zero during each power cycle. This behavior confirms that the device is operating in DCM, as expected.

The two large capacitors — C1 and C2 — are used to help reduce the ripple on the output voltage. Any switching supply, no matter how well regulated and filtered, has some ripple (also known as output noise). Without the large capacitors, the A/D conversion is unsteady and the output voltage tends to oscillate. The output ripple is not visible in the output voltage plot in Figure 7, but a close examination of the simulation reveals that it is there.

In addition to large output capacitors, ripple can be further reduced by the addition of output filters. An output filter consists of a small inductor L (2-10 µH) and a capacitor C1 (50-150 µF) in parallel with the

■ FIGURE 8. Boost power supply output filter.



REFERENCES

1. Boost Steady State Discontinuous Mode Analysis, downloaded 11/14/2005 from **http://focus.ti.com/lit/an/slva061/slva061.pdf**

2. *Nuts &Volts* FTP site at **www.nutsvolts.com/ftpindex.htm**

3. A. Williams, "Spice Up Your PC," *Nuts & Volts*, July 2002, pp. 10-12.

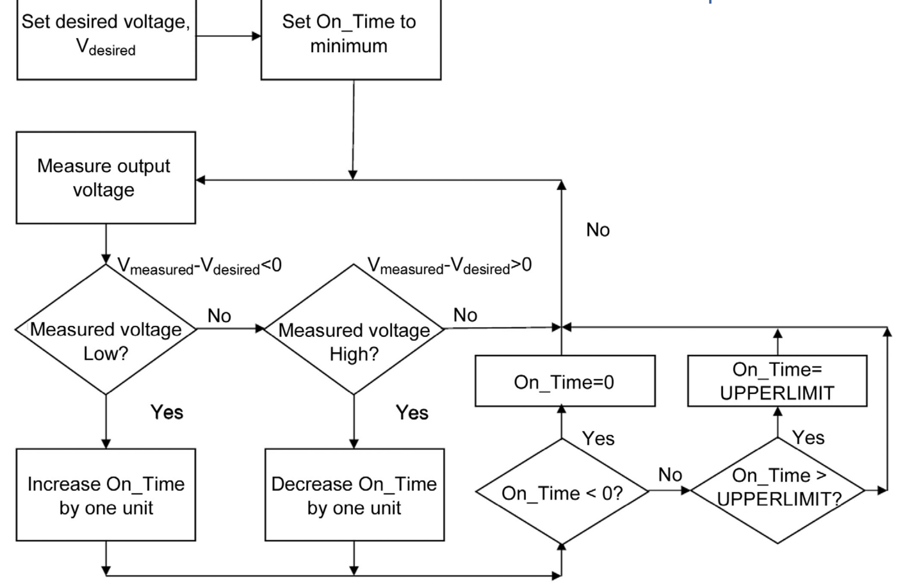
4. LTSPICE Version 2.16h free 5.4 MB download retrieved 10/5/2005 from **www.linear.com/designtools/softwareRegistration.jsp**

5. IRF730 SPICE model retrieved 10/6/2005 from **www.irf.com/product-info/models/spice.**

6. **MBRS3201T3** SPICE model retrieved on 11/9/2005 from **www.onsemi.com/PowerSolutions/product.do?id=MBRS3201**

7. R. Condit and D. Butler, "Low Cost Microcontroller Programmer," Microchip App Note 258, retrieved 10/21/2005 from **www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1999&ty=&dty=&section=&NextRow=&ssUserText=an258&DesignDocSelect=&x=7&y=9**

■ FIGURE 9. Simplified flow chart.



output capacitor as shown in Figure 8. The inductor must be rated at full load current since the load is connected across the output filter capacitor.

Figure 9 shows the basic PWM logic implemented on the microprocessor. Reference 7 has a more complicated algorithm that involves looking not only at the voltage error, but also at the integral of the voltage error and the derivative of the voltage error. This more complicated correction algorithm is intended to reduce voltage oscillations and improve convergence of the output voltage to the desired voltage. You may want to try it if your output voltage oscillates or fails to converge to the desired voltage.

Conclusion

This article covered the theory of how a boost power supply works. It described pulse width modulation and explained how to get the 18F2455 PIC to output a square wave with a given period and ON time. It covered the important parameters in designing the boost power supply, showed how to use the free LTSPICE program to come up with a possible design, and looked at the output of a simulation using this design.

Next month, we will build the power supply we just designed. We will write the software needed to drive the PIC in a freely downloadable version of the C language. We will discuss the software that drives the A/D conversion, the LCD display, and the PWM. Then we will program the microprocessor with the software. I will mention some of the pitfalls in designing a switching power supply and, finally, use the power supply to power some simple applications. **NV**

■ BY L. PAUL VERHAGE

NEAR SPACE BOOMS AND SOUNDS

THIS MONTH, WE'LL LOOK AT two near space experiments. The first one is an engineering test and the second a traditional science experiment.

A TEST OF COMPOSITE BOOM STRENGTH

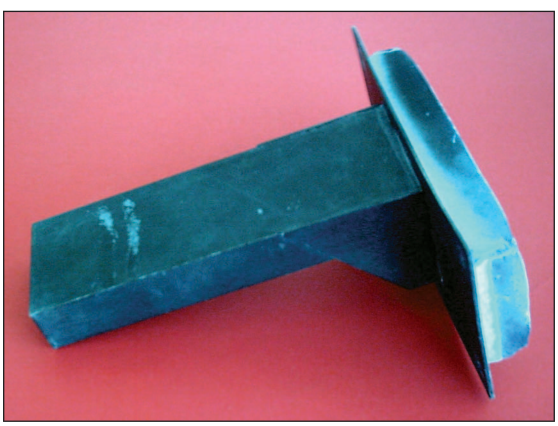
In Chapter 5, Section 3.3 of my book *Near Space Exploration with the BASIC Stamp* (which is available online as a free download from Parallax at **www.parallax.com**), you'll find directions for making strong, stiff, and lightweight composite booms for near spacecraft. The booms, which are mounted to quad panels, let me quickly switch out antennas and experiments on a near spacecraft. After testing the strength of different types of Styrofoam for my BalloonSat article, I wanted to test the strength of my composite booms. Before describing the test and its results, let me briefly explain how I make booms. I believe you'll find these composite booms are useful not just for near space, but also for robotics.

MAKING A BOOM

I lucked out. Back in 1997, I stumbled upon how to make these booms. I had just completed my second near spacecraft airframe and needed a boom to mount experiments and antennas away from the airframe. As a test, I cut a Styrofoam sheet to the dimensions of the boom I wanted and laminated four of its sides with thin modeling plywood (I got the idea from a 1980's *Sky and Telescope* article about laminating Styrofoam in fiberglass). The results were so outstanding that I've been using composite booms ever since.

The Styrofoam core of my booms comes in four by eight foot sheets and is available from a local home improvement store. I don't use the soft white Styrofoam of cheap ice coolers because it has large grains that don't adhere well to each other.

The laminating plywood is sold by a hobby store in town. The plywood is 1/32 inch thick and I usually purchase it in two by four foot sheets. While at the hobby store, I also purchase epoxy in four ounce bottles. I don't recommend the tiny tubes of epoxy sold at hardware stores. You'll need a lot of epoxy to make booms and, in those quantities, the larger bottle of hobby store epoxy is cheaper. Besides, I suspect the hobby store epoxy is better quality. Now that you have the parts, let's make a boom.



■ FIGURE 1. A completed boom, painted and mounted to a quad panel. It's ready to rock and roll.

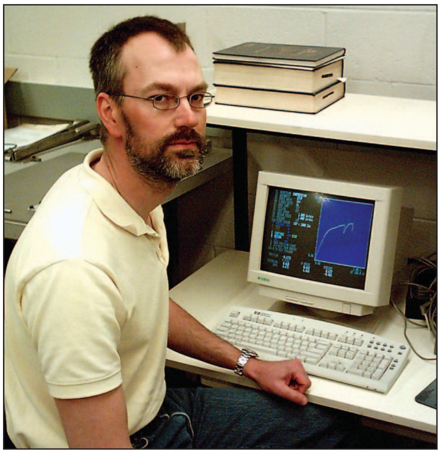
Begin by cutting the Styrofoam to size. Next cut four pieces of 1/32-inch thick plywood a little larger then the four largest faces of the Styrofoam core. Then laminate the first two pieces of plywood to opposite sides of the Styrofoam. To limit the mess, mix the epoxy right on the face of the Styrofoam and spread it thinly with a Popsicle stick. After gluing the first two faces on the Styrofoam, wrap several pieces of masking tape around the boom to hold the plywood in place while the epoxy sets.

After the epoxy sets, remove the masking tape and sand the two unlaminated sides of the boom until the edges of the plywood are flush with the Styrofoam. A stationary belt sander is a great tool for doing this as it quickly creates flat surfaces on the boom. After sanding the bare sides of the boom, apply epoxy to the two remaining bare sides of the boom and cover them in plywood. Again wrap the boom in masking tape to hold the plywood in place until the epoxy sets.

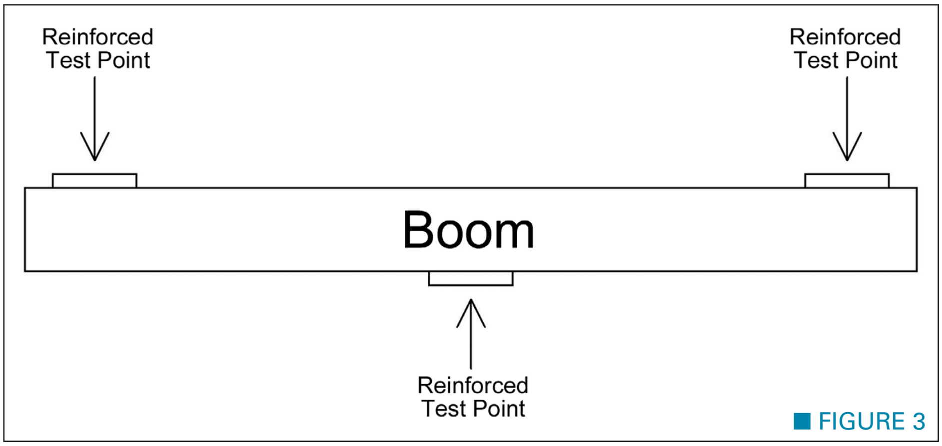
After the second batch of epoxy sets, sand the boom one last time to clean up the two new plywood edges. I cover one of the small ends of the boom in plywood, but the exact dimensions and shape depend on the boom's function. The remaining end is epoxied to a quad panel, so I don't laminate it like the other sides. I like to finish my booms by giving them a coat of spray paint to seal out the weather and make them look nice.

TESTING MY BOOMS

In my previous column on BalloonSats, I tested the strength and stiffness of both 1/4" thick foam core and 1/2" thick Styrofoam by hanging



■ FIGURE 2. At the press and getting ready to test some booms. Without the help of Dr. Anthony Paris, I couldn't have written this article.



■ FIGURE 3

weights off of identically cut samples and measuring their deflection under load. While this worked well for the Styrofoam test, I realized it wouldn't work for the much stronger composite booms. So I contacted the Mechanical Engineering department at Boise State University (**http://coen.boisestate.edu/me**) for help. Dr. Anthony Paris was kind enough to take a look at my project and test the booms.

I made six booms using different laminating materials and configurations for this article. Each boom was eight inches long, 3/4 inches high, and one inch wide. To keep the test equipment from digging into the booms, I glued 1/8-inch thick panels over the three test points on each boom as illustrated in Figure 3.

Here are the boom designs and weights:

• Bare Styrofoam boom with no covering on any surface (two grams)

• Boom wrapped in tape used to cover Styrofoam gliders (three grams)

• Boom covered with 1/32" thick plywood on left and right sides (12 grams)

• Boom covered with 1/32" thick plywood on top and bottom sides (14 grams)

• Boom covered on all four sides with 1/32" thick plywood (20 grams)

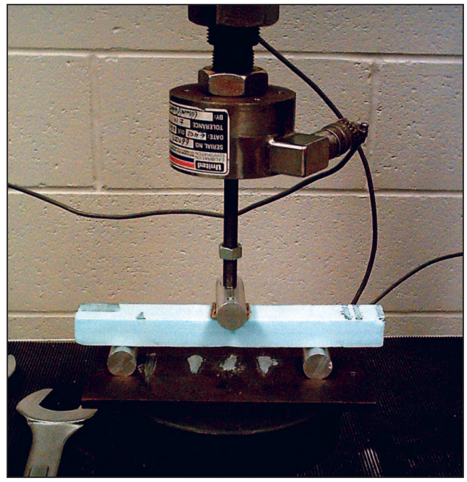
• Boom covered on all four sides with 1/16" thick plywood (27 grams)

A PC-controlled press was used to test the booms. Software operates the stepper motor in the press, pushing a ram against the test sample. At the end of the ram is a load cell. Load cells indicate the force acting on them by producing a voltage that is proportional to the force.

It took several minutes to run each test. For protection, we wore safety glasses in case the plywood shattered under load.

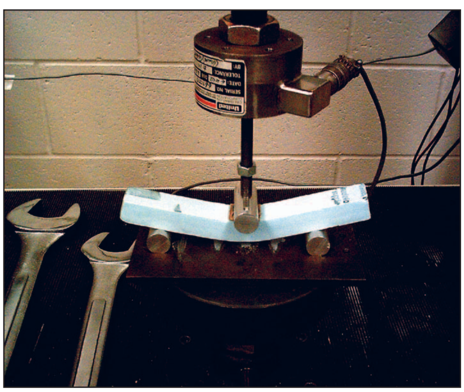
During the test, the PC recorded the displacement of the ram and the force measured by the load cell. The data generated was stored in a comma delimited text file and was formatted like the sample below:

```
"LOAD","STRAIN"
pounds,inches
 0 , 0
-3.433228E-02 , 0
-2.670288E-02 , .0001
-2.670288E-02 , .000225
-2.288818E-02 , .000425
```



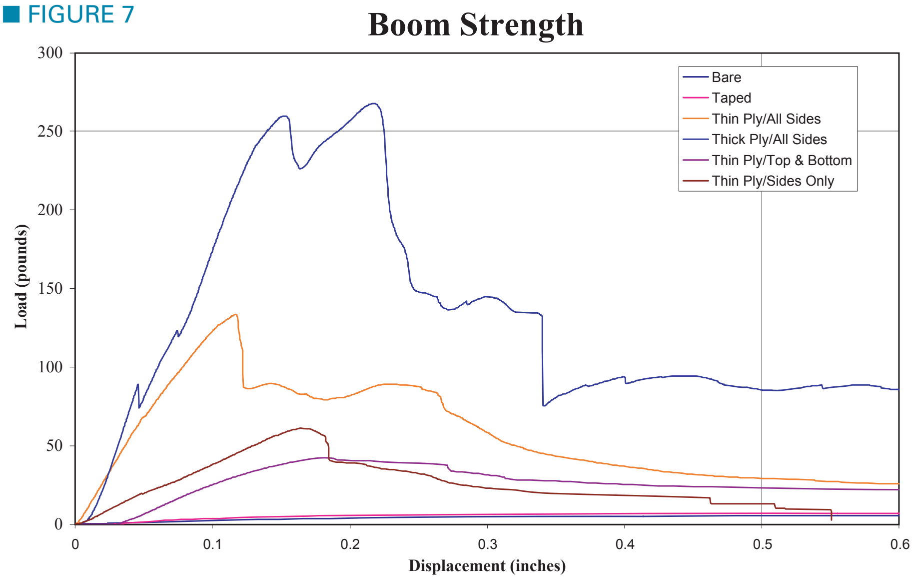
■ FIGURE 4. A boom about to undergo testing. The boom is supported on two metal rollers and the ram pushes down on a third roller on top of the boom. The load cell is the thick disk near the top of the image.

■ FIGURE 6. The plywood lamination in this test helped the Styrofoam boom resist bending. But when it gave out, the boom broke rather than bent (the broken plywood is more visible on its underside).



■ FIGURE 5. Note that the bare boom is bending without fracturing.

■ FIGURE 7



The first column is the amount of force measured by the load cell and the second column is the displacement of the ram in the press. Around 2,000 measurements were generated during each test.

I imported the results into an Excel spreadsheet and generated a chart shown in Figure 7. If you're interested in this data, you can find a copy on the *Nuts & Volts* website (**www.nutsvolts.com**).

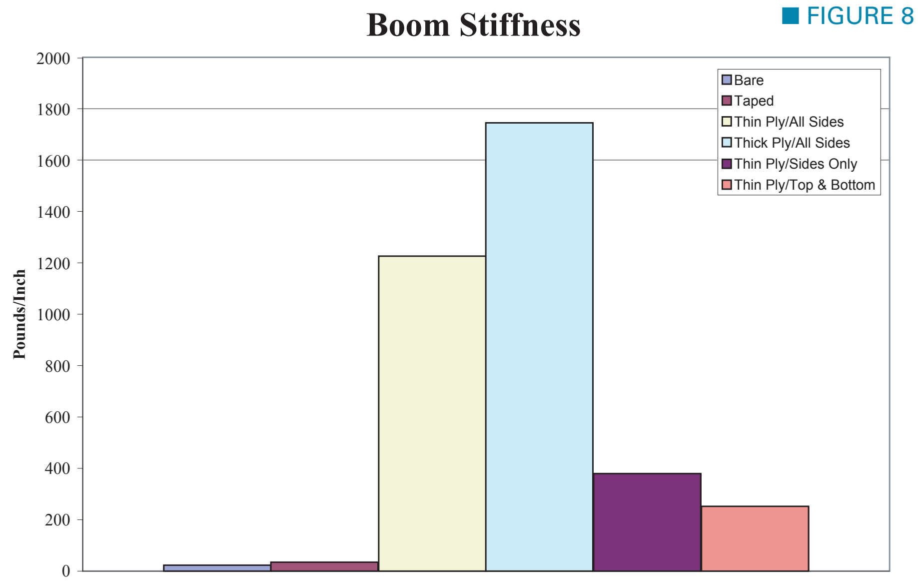
ANALYSIS

A force of only 10 pounds will flex a bare 3/4-inch thick Styrofoam boom by its own thickness. A taped boom does so with a force of 11 pounds. Look at what plywood lamination does to Styrofoam, however. Booms laminated on two sides can withstand 40 pounds of force without failing and their deflection is minimal until they fail. Booms laminated on all four sides can withstand forces exceeding 100 pounds with minimal deflection. A 3/4-inch thick Styrofoam boom eight inches long and covered in 1/16" thick plywood can support my weight and hardly bend. And that's a boom that only weighs 27 grams, or one ounce.

After looking over the data, Dr. Paris explained that the slope of each curve represents the stiffness of the materials. I had my spreadsheet calculate the average slope of each curve between the 0.0 and 0.1 inch deflection points. The chart in Figure 8 shows the stiffness of each of the booms.

Stiffness is measured in units of pounds per inch (or newtons per meter in the metric system). The chart above shows that a fully laminated boom is at least 500 times stiffer than a bare Styrofoam boom. It's interesting to note that a Styrofoam boom laminated in 1/16" thick plywood has a stiffness close to one ton per inch.

■ FIGURE 8



CONCLUSION

So, what have I learned? If the amount of force that a boom must support is small (less than 10 pounds), then a bare Styrofoam boom can handle it and bounce back after deflecting. However, if the force acting on a boom is on the order of 100 pounds or more, then a fully laminated Styrofoam boom is required if the boom is not to break. If I can live with a boom 15% heavier than a 1/32" covered boom, then I can make a boom capable of supporting up to 250 pounds with 1/16-inch thick plywood.

The stiffness of a fully laminated boom is significantly greater (it flexes less) than a bare or even partially laminated boom. A fully laminated Styrofoam boom under a load of 100 pounds experiences less than 0.1 inches of deflection. That's not bad for something you can make in your garage.

If you have a near spacecraft or

■ FIGURE 9. Here I am at the press.

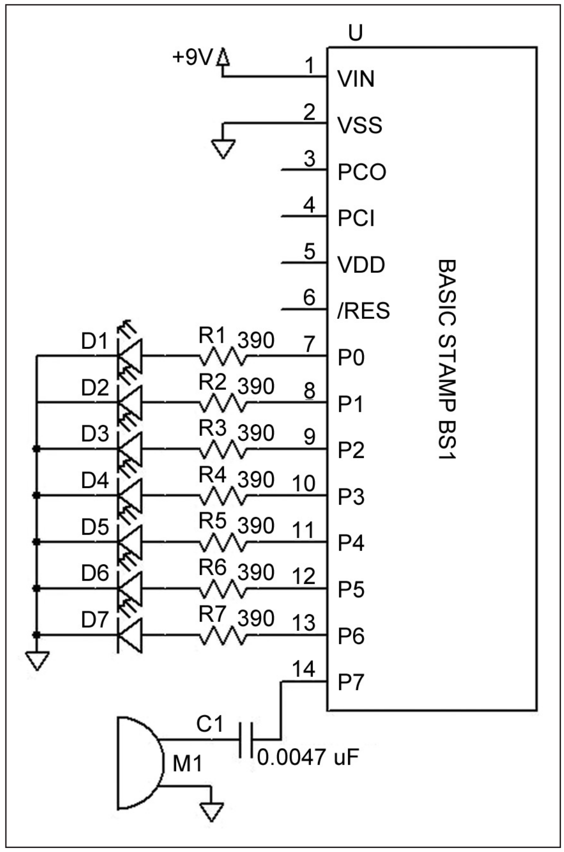
robot that needs a stiff and strong structure that is lightweight and cheap, then consider making a composite structure from Styrofoam and thin plywood.

SOUNDS OF NEAR SPACE

Now let's switch gears from ground testing to a near space experiment. In the winter of 1997, I launched what was probably the first camcorder into near space. The scenery was pretty spectacular. But along with the change of scenery, I noticed a hush slowly developing as the near spacecraft climbed higher. The camcorder's noises were loud at launch but impossible to hear at 85,000 feet. I knew sound could not travel through the vacuum of space, but I didn't know how sound was affected as the pressure dropped. For example, does sound gradually fade with decreasing pressure or drop off abruptly? How do different frequencies react as the pressure decreases?

As an end-of-semester project, I had some students in my high school electronics class design an audio experiment for near space. Students in my class learn to build circuits and program the BASIC Stamp. I had them use a Parallax application note to build a tone generator. Knowing that at some point it would be impossible to hear the tones, I had my students incorporate LEDs into the circuit to indicate which tone was being generated.

To gather the data, the experiment was attached to the near spacecraft airframe (with a boom) and recorded with a camcorder. The BASIC Stamp 1 (rev D.) controlling the experiment cycled through the tones in one minute intervals as the balloon rose to an estimated altitude of 85,000 feet. After recovering the near spacecraft — which took five months due to an onboard tracker failure — I played back the video tape. It was obvious the tones were not equally affected during the ascent. Higher frequency tones grew quieter sooner than the low frequency tones. (The tape sat untouched until this year when I found a way to analyze its data.)



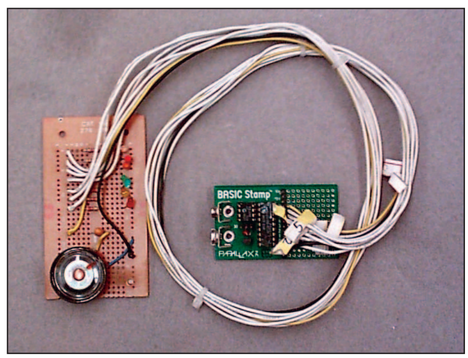
■ FIGURE 10. The schematic my students used to make the BASIC Stamp tone generator.

GREAT RESULTS, NOW LET'S ANALYZE IT

This spring, I visited Dr. Joe Guarino at the College of Engineering at Boise State University. Dr. Guarino's specialty is acoustical engineering. He analyzes sounds on a PC with a sound card and software called Spectra Pro. To analyze the tone data on my camcorder tape, we connected the audio output of a VCR deck to the inputs of the PC's sound card. As we played the VCR tape from the mission, Spectra Pro software measured each tone's frequency and volume.

First, it was necessary to measure the frequencies of the tones generated by the BS-1 since I no longer had a copy of the program my students wrote. Next, we measured the volume of each tone at specific intervals. Since there wasn't a GPS log, I didn't know the altitude of the experiment.

The typical ascent rate for my near spacecraft back then was around 1,000 feet per minute. So from the VCR's displayed time, I estimated the



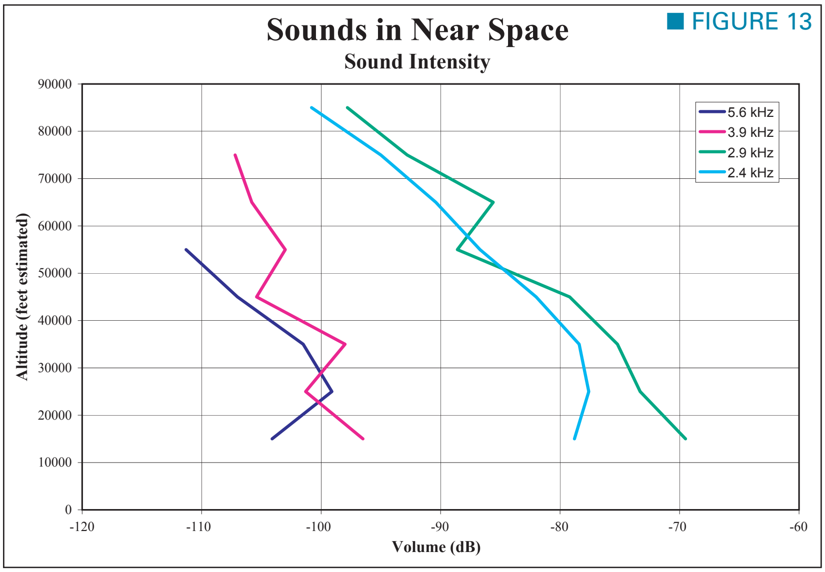
■ FIGURE 11. The completed experiment. For sitting out in the wild for five months, it's none worse for wear.

SPECTRUM ANALYZER DATA

Altitude (estimated feet)	Channel Volumes (dB) 1	2	3	4	5	6	7
15,000	X	-104.1	-96.5	-69.5	-78.8	X	X
25,000	X	-99.1	-101.3	-73.3	-77.6	-108.8	-88.5
35,000	X	-101.5	-98.0	-75.2	-78.4	-99.7	X
45,000	X	-107.0	-105.4	-79.2	-82.0	X	X
55,000	X	-111.3	-103.0	-88.6	-86.7	X	X
65,000	X	X	-105.8	-85.6	-90.4	X	X
75,000	X	X	-107.2	-92.8	-95.0	X	X
85,000	X	X	X	-97.8	-100.8		

■ FIGURE 12. Spectrum analyzer data. The altitude is estimated based on a 1,000 foot per minute ascent rate. An X indicates no meaningful data could be collected.

FIGURE 13



played. Several times we backed up the tape and try clicking again. Even when we clicked the mouse during a tone, we often did not measure the tone at its loudest volume.

THE DATA COLLECTED

From measurements taken at ground level, we determined the seven tones had these frequencies:

Channel 1	10.5 kHz
Channel 2	5.6 kHz
Channel 3	3.9 kHz
Channel 4	2.9 kHz
Channel 5	2.4 kHz
Channel 6	2.2 kHz
Channel 7	0.5 kHz

altitude of the balloon each time we measured the tone volumes. I decided to measure each tone every 10,000 feet or at 10 minute intervals. This limited the amount of data collected, but also limited the burden I was placing on Dr. Guarino. One of the problems we experienced was that Spectra Pro only measures sound volume at the instant the mouse is clicked. So if our timing was off — and often it was — we didn't get the volume of the tone while it was being

The noise floor for the video tape was -125 dB. This means that when there were no tones on the tape, the background noise had a volume of -125 dB. This also means that tone volumes couldn't be measured once they fell below -125 dB.

ANALYSIS

From the information above, I generated the a chart with Excel (shown in Figure 13).

First, notice that the volume of each frequency decreases with increasing altitude. Also notice that the tones are not equally affected. You can see that the lower the frequency of a tone, the higher the altitude it could be recorded by the camcorder. I was expecting smooth changes in volume and suspect my data would be smoother if we timed our sampling better.

What does this data mean and what influences the maximum altitude that a tone can be heard at? I don't know. I've charted the speed of sound and the mean free path of molecules as a function of altitude, but I can't find a correlation. Perhaps if I sampled the data better, a relationship would become apparent. Like many good science experiments, I'm left with more questions than answers. So, perhaps it's time to redesign this experiment and fly it again.

IMPROVEMENTS FOR NEXT TIME

Here's how I plan to improve the design of this experiment before its next flight. First, I'll create a better tone generator. To do this experiment right, I need a wider range of frequencies. The one inch diameter speaker used in the original experiment was ineffective at generating low frequency sounds. So I'll use larger speakers for lower frequency tones and smaller speakers for higher frequency tones.

Second, I need to lower the noise level. I plan to use Styrofoam to build an enclosure that blocks background noise and shields the tone generator and sound recorder

from balloon and ground noise. A near spacecraft makes a surprising amount of noise as it ascends.

Third, instead of using a noisy camcorder to record sounds, I'll use my Fidelity DV digital media recorder. The Fidelity DV stores mpeg files on a Flash card. After the flight, I'll attach the Flash card to my PC as a drive. There will be no need to use audio cables or to digitize sounds from the recording.

Finally, I'll use the Audacity program to analyze the tones. Audacity lets me load the sound recording as a project and step through the entire sound file in steps of a fraction of a second. This is going to make spectrum analysis of the recorded tones much easier.

BLATANT SELF PROMOTION

Where did I find out about Audacity? Well, I'm the producer for a radio program on astronomy called Idaho Skies. To edit the show's recording, the radio station taught me to use Audacity. You can download this software for free from Source Forge at, **http://audacity.sourceforge.net/**.

If your PC has a sound card, you can use Audacity to create audio shows at home. Audacity is very simple to use and packed with features (like the spectrum analyzer). By the way, if you're interested in space and astronomy, then listen to Idaho Skies on Tuesday, Wednesday, and Thursday at 12:55 PM (Mountain Standard Time). Idaho Skies is a webcast of the Boise Community Radio Project. They can be found at radioboise.org.

Onwards and Upwards,
Your near space guide **NV**

■ FIGURE 14. My Fidelity digital camcorder. While the images are only 240 x 260, I can get audio and video for an entire flight on a 1 Gb SD RAM. Best of all, it only weighs seven ounces.

■ FIGURE 15. Beauties and the Beast. I couldn't produce Idaho Skies without the talents of these two young ladies, Kris and Rachel. Aren't I lucky?

■ BY PETER BEST

THE LAND OF TCP/IP

IN A PREVIOUS DESIGN CYCLE (July '06), we examined what I consider to be the least complex of all of the Internet protocols — UDP. In that segment of Design Cycle, I also presented some LAN-ready AVR hardware and associated C source code to enable you to use the UDP protocol in your own embedded projects.

In this edition of Design Cycle, we're going to march cross-country into the land of TCP/IP. Although the same hardware used to transmit and receive UDP datagrams can be used to transport TCP/IP packets, TCP/IP is a bit more complex to code than UDP. However, that's not going to stop us from getting a microcontroller version of TCP/IP up on a LAN.

Complex things become simple when broken down into component parts, and that's exactly what we're going to do with TCP/IP to gain an understanding of how the TCP/IP protocol works. Once you've got a grip on TCP/IP basics, I'll introduce you to some new embedded hardware called the Frame Thrower II that we'll use to run our minimal TCP/IP stack.

TCP/IP 101

As you can see in Figure 1, encapsulation is used to transport TCP segments within an IP datagram in the same way that encapsulation is employed by UDP datagrams.

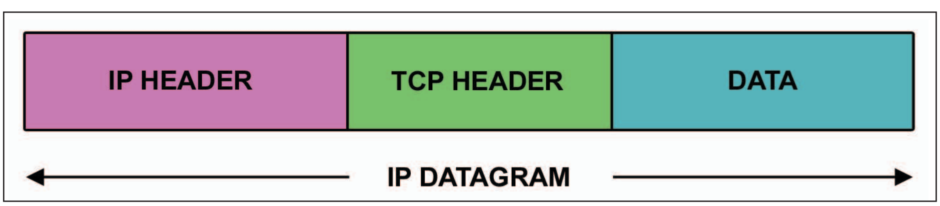
If you look at the TCP segment from an Ethernet point-of-view, the Ethernet type field identifies the frame inside its data area as an IP frame. If the value in the protocol field of the IP header is equal to 6, the packet payload is a TCP segment. To better illustrate this concept, I've captured an Ethernet encapsulated IP header of a TCP segment in Screen Shot 1.

TCP/IP was designed to be platform independent. If you've ever had the opportunity to loiter in computer rooms, you know that TCP/IP can be run on tiny microcontroller-based systems, personal computers, or large mainframe complexes. TCP/IP consists of a collection of protocols that are standardized in the networking community. The most common way to implement TCP/IP is via a "TCP/IP stack." Complete TCP/IP stacks can be very large. Usually, a fully compliant stack can't be easily ported to small, memory-constrained embedded systems like the Frame Thrower II.

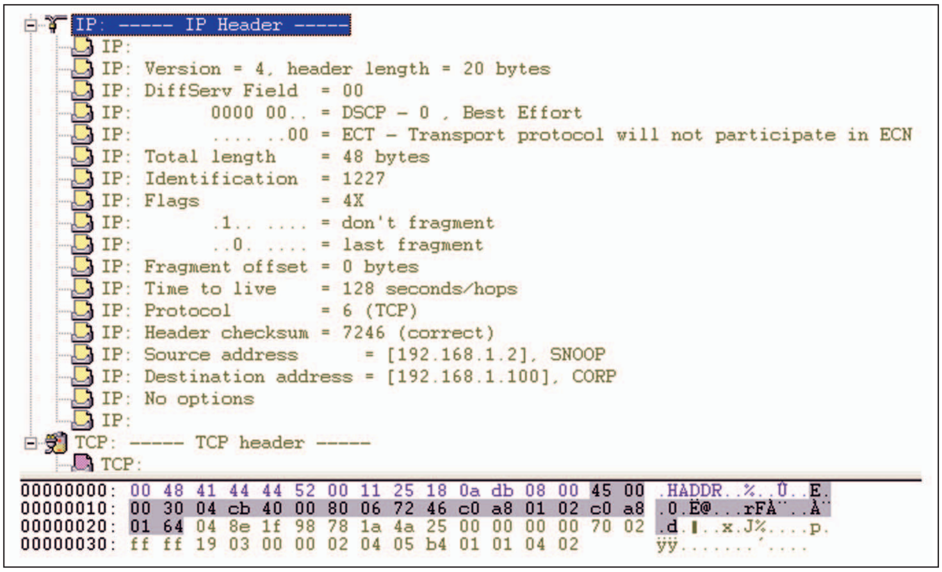
Just as we saw with UDP network devices, every machine or device on a TCP/IP-based network — no matter how big or how small — is called a host. That includes clients, as well as servers. It used to be that a server was always the big machine in the cloud and the client was a workstation on someone's desk. Today, servers can sit on desktops and servers as well, as clients are becoming more and more microcontroller-based. Regardless of the host's size, the idea is that all hosts can communicate with each other. This implies that all hosts on all networks can communicate host to host across differing networks. This may sound impractical from a hardware standpoint, but that's how the Internet really works.

TCP/IP messages are generally short packets of data. Just like UDP, each TCP data packet is addressed to reach a particular host on a particular network. There is no difference in the addressing scheme used for TCP and UDP. The IP address for both protocols is the familiar four bytes divided by dots address scheme (192.168.1.100). Six-byte physical addresses or MAC addresses apply to both UDP and TCP and are used in the identical manner by both protocols.

You're probably used to hearing the term "TCP/IP" and thinking about



■ FIGURE 1. Encapsulation is what this is all about. The IP and TCP headers are very important parts of the game as they hold valuable addressing information, as well as information about the data contained within the packet.



■ SCREEN SHOT 1. This is a Sniffer capture of a real TCP/IP packet issued by my laptop aimed at the Frame Thrower II. Note the IP addresses and the declaration of the TCP protocol. The associated bytes that go with the decoded text are highlighted in the dump area at the bottom of the shot.

it as one protocol. In reality, IP is its own protocol just as TCP is. Just as it is in the UDP world, IP is the unreliable component of the TCP/IP pair. IP is termed unreliable because there is no way that IP itself can guarantee that a data packet will actually be delivered to its destination. All of the hosts in the Internet or on a local LAN give their best effort to deliver an IP data packet.

The TCP/IP protocol stack is composed of five layers. The Physical layer is the simplest layer with the Application layer normally being the most complex of the five. Each layer of the TCP/IP stack has a distinct job to do.

The Physical layer is the wire, cable, and electronics that connect the devices and networks to each other. The Physical layer always sees the entire packet — whether it is receiving it or transmitting it — and never adds or subtracts to a packet's contents. In our application, the Ethernet cable, the Ethernet isolation magnetics, and the interface circuitry contained within the Frame Thrower II's Ethernet engine IC form the Physical layer.

The Data Link Layer is responsible for transferring a datagram from one host over a physical link to another host. Data Link Layer functionality resides in the Ethernet engine IC's MAC engine. The MAC engine accepts data and wraps it into an Ethernet-compatible package that can be received and unwrapped by the host the data was addressed to. If we look at the MAC engine from an Ethernet point-of-view, the MAC engine is responsible for generating the Ethernet frame preamble and CRC.

The MAC engine also makes sure the ether is clear before attempting to send a message and if a collision occurs, the MAC engine is programmed to retry the transmission. One of the most important things the MAC engine does is check the incoming frame's hardware address to determine if the incoming data belongs to it or another host on the network.

The Network Layer encapsulates messages passed from the Transport Layer and produces datagrams. IP lives in the Network Layer. The Network Layer encapsulates a UDP packet or TCP segment inside an IP datagram. An IP header is added and the datagram is then passed along to the Data Link Layer for transmission.

The Transport Layer lies between the Application Layer and the Network Layer. The Transport Layer passes data between the Application Layer and the Transport Layer using TCP or UDP protocols. UDP and TCP live in the Transport Layer. TCP, unlike UDP — which is a connectionless protocol — uses a virtual connection to make sure that the data arrives at its destination intact and in order. TCP accomplishes this virtual connection via handshaking and special codes in each TCP segment. TCP and UDP receive data from the application and form segments or packets, respectively. A destination address is added before passing the packet or segment to the Network Layer.

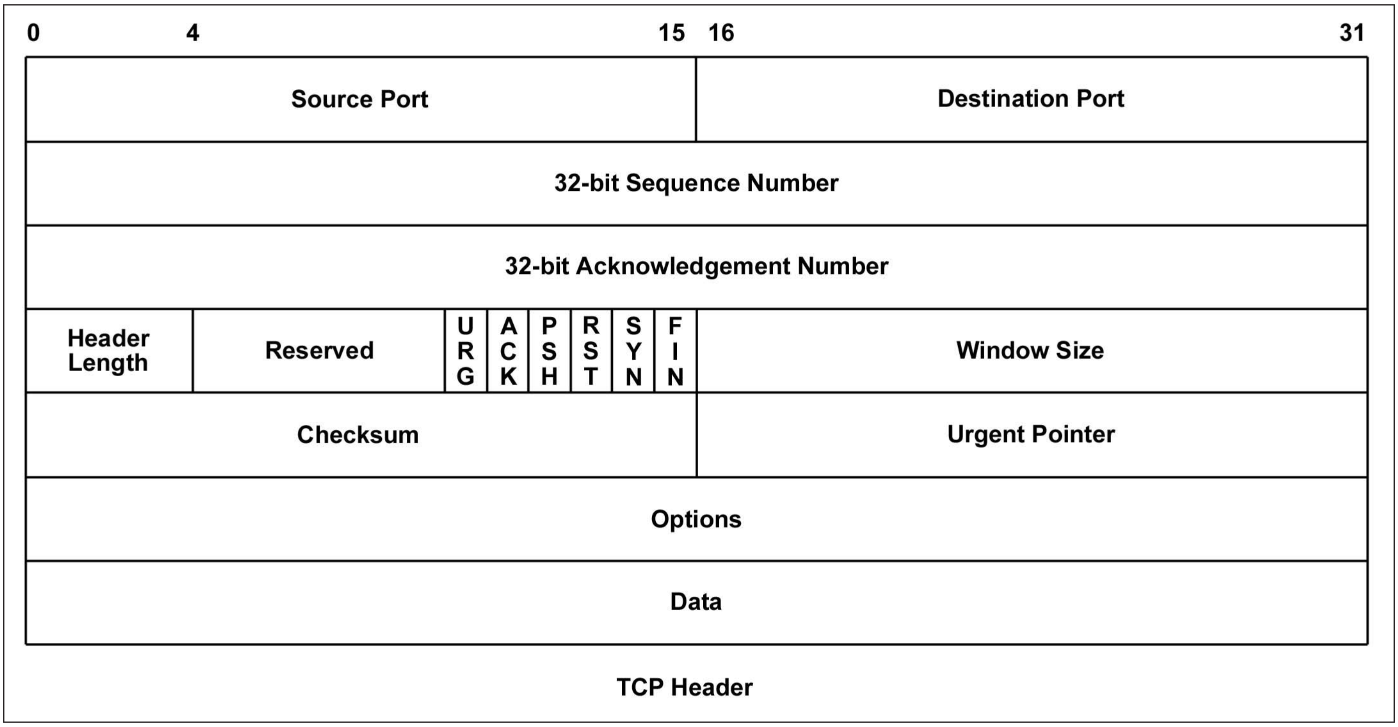
The Application Layer is the domain of the programmer. Just about anything goes in the Application Layer. Data flows from the Application Layer of the originating host down through the TCP/IP stack and out the Physical Layer across to the Physical Layer of the destination host. Once the data enters the destination host's Physical Layer, the TCP/IP stack's layer process is reversed and the data flows up through the TCP/IP stack to the Application Layer where it is processed.

Despite all of the talents each layer of the TCP/IP stack possesses, encapsulation is the glue that holds the TCP/IP stack together. Each stack layer passes only properly formatted output to the next stack layer for processing. Encapsulation also allows each layer to treat the data in the way it prefers without affecting the way the data is treated in other layers.

Now that you have knowledge concerning the components of a TCP/IP stack, let's turn our attention to the TCP header. I've posted a TCP header layout graphic in Figure 2.

The TCP header lies at the beginning of the IP data area. The IP data area is located just beyond the source and destination IP addresses in the IP header. If you take another peek at Screen Shot 1, you'll see the *TCP header* area is next up after the source and destination IP address information.

A complete TCP header can be seen in Screen Shot 2. Flip back and

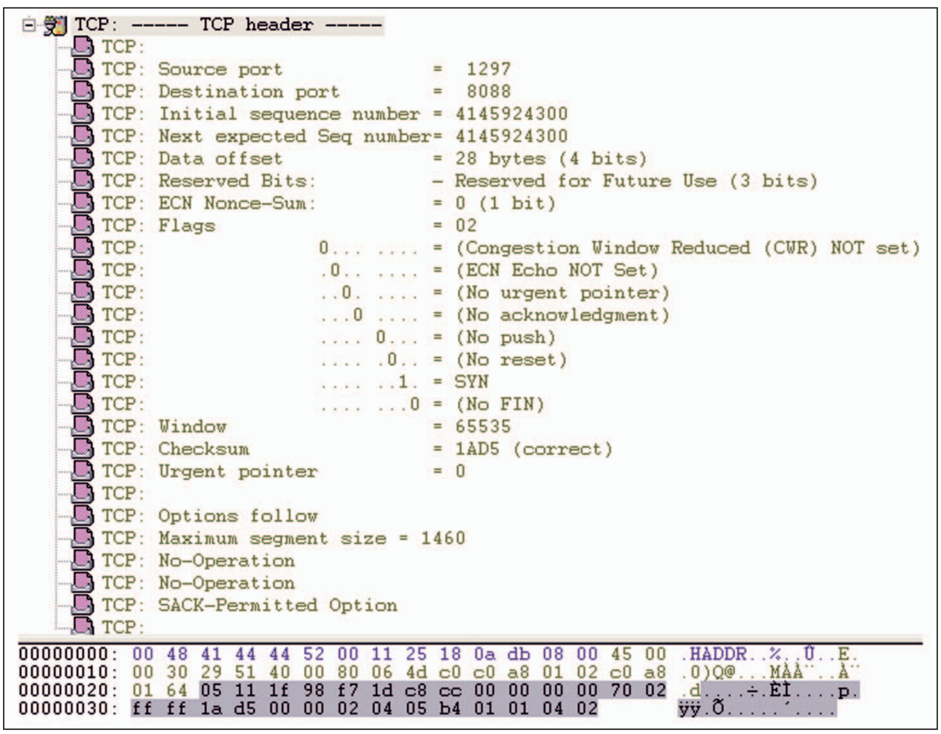


■ FIGURE 2. This isn't as ugly as it may seem. As we go along, I'll use sniffer captures to explain what is going on in each of the areas of this graphic that we're interested in. The sequence and acknowledgement numbers, along with the flags, make the TCP segment unique with respect to the UDP datagram.

forth between Screen Shot 2 and Figure 2. You'll see that you can physically match the sniffer fields with the fields delineated in the Figure 2 layout graphic. The first 16 bits of the TCP header contain the TCP source port address. The destination port resides on our embedded TCP/IP host, the Frame Thrower II. The TCP/IP embedded host firmware defines the Frame Thrower II's TCP port as 8088 decimal. 1297 decimal is the TCP port being used by the Telnet application on the laptop computer I used to generate the TCP/IP data stream.

What you see in Screen Shot 2 is the beginning of what is called the three-way handshake. The first message sent in the three-way handshake process is a dataless TCP segment with the SYN bit set in the TCP header flag's field. If there is no TCP session established, the Frame Thrower II's first inclination is to check the destination port address and, if the destination port belongs to the Frame Thrower II, it then runs a test on the SYN flag bit in the TCP header.

If it is determined that the remote host (my laptop) is trying to establish a TCP/IP session (in this case, it is), the receiving host (the Frame Thrower II) prepares for part two of the three-way handshake by incrementing the incoming sequence number from the remote host before sending it back as the acknowledgement number.



■ SCREEN SHOT 2. You can easily tell who's talking and who's listening by associating the source and destination port numbers with their respective hardware. In our case, port 8088 belongs to the Frame Thrower II and port 1297 resides at the laptop end of the link. This capture is the first shot fired in the TCP three-way handshake process.

■ SCREEN SHOT 3. Take a look at the initial sequence number in Screen Shot 2. Note that the acknowledgement number in this capture is actually the initial sequence number in Screen Shot 2 incremented by one. This is one of the ways TCP keeps up with the amount of data passed in a TCP segment. This capture is the second shot fired in the TCP three-way handshake process.

```
TCP: ----- TCP header -----
  TCP:
  TCP: Source port              =  8088
  TCP: Destination port         =  1297
  TCP: Initial sequence number  = 196607
  TCP: Next expected Seq number= 196607
  TCP: Acknowledgment number    = 4145924301
  TCP: Data offset              = 28 bytes (4 bits)
  TCP: Reserved Bits:           - Reserved for Future Use (3 bits)
  TCP: ECN Nonce-Sum:           = 0 (1 bit)
  TCP: Flags                    = 12
  TCP:                0... .... = (Congestion Window Reduced (CWR) NOT set)
  TCP:                .0.. .... = (ECN Echo NOT Set)
  TCP:                ..0. .... = (No urgent pointer)
  TCP:                ...1 .... = Acknowledgment
  TCP:                .... 0... = (No push)
  TCP:                .... .0.. = (No reset)
  TCP:                .... ..1. = SYN
  TCP:                .... ...0 = (No FIN)
  TCP: Window                   = 65535
  TCP: Checksum                 = 1AC2 (correct)
  TCP: Urgent pointer           = 0
  TCP:
  TCP: Options follow
  TCP: Maximum segment size = 1460
  TCP: No-Operation
  TCP: No-Operation
  TCP: SACK-Permitted Option
00000000: 00 11 25 18 0a db 00 48 41 44 44 52 08 00 45 00  ..%..Û.HADDR..E.
00000010: 00 30 29 51 40 00 80 06 4d c0 c0 a8 01 64 c0 a8  .0)Q@...MÀÀ¨.dÀ¨
00000020: 01 02 1f 98 05 11 00 02 ff ff f7 1d c8 cd 70 12  ........ÿÿ÷.ÈÍp.
00000030: ff ff 1a c2 00 00 02 04 05 b4 01 01 04 02 a2 78  ÿÿ.Â......´....¢x
00000040: d7 79 41                                         ×yA
```

The SYN flag in the incoming frame of the laptop's first TCP handshake segment must be treated as a sequence number in the Frame Thrower II's response, which is the second part of the three-way handshake. The laptop is expecting the value of the initial sequence number field to be incremented by one, as shown in the Frame Thrower II's response captured in Screen Shot 3.

Before sending an acknowledgement, the Frame Thrower II must establish an initial sequence number (ISN) of its own. The ISN (Initial Sequence Number) is a 32-bit pseudo random number that is ultimately placed into the Frame Thrower II's outgoing TCP sequence number header field. During this run, the Frame Thrower II's ISN is set for 196607. In this part of the three-way handshake process, the Frame Thrower II must respond with the SYN and ACK flag bits set in the TCP header. You can see all of the SYN/ACK action in the Flags portion of Screen Shot 3.

Now that all of the sequence numbers have been exchanged, the only thing standing between the Frame Thrower II and the laptop's exchange of real data is the final acknowledgement from the laptop. The final ACK TCP segment is captured and displayed in Screen Shot 4. The three-way handshake process is now complete and the laptop and Frame Thrower II are ready to exchange data.

The laptop has gotten to this point using a Telnet session. If it weren't for the sniffer screen shots, the human behind the Telnet wheel would not know if anything has happened between the laptop and the Frame Thrower II. So, to throw the human operator a bone, the Frame Thrower II follows the three-way handshake with a banner message. The Telnet banner message is hard-coded into the Flash of the PIC18F8722, which is in charge of the Frame Thrower II's operations. If you look closely in the hexadecimal dump area of Screen Shot 5, you'll see the Telnet banner message follows a carriage return/line feed

■ SCREEN SHOT 4. This capture is particularly interesting in that the laptop incorrectly calculated the ACK segment checksum. Note the change in the sequence number and the acknowledgement number. As more data flows, these numbers change in direct proportion to the amount of data passed between the Frame Thrower II and the laptop. This is the final shot of the three-way handshake process.

```
TCP: ----- TCP header -----
  TCP:
  TCP: Source port              =  1297
  TCP: Destination port         =  8088
  TCP: Sequence number          = 4145924301
  TCP: Next expected Seq number= 4145924301
  TCP: Acknowledgment number    = 196608
  TCP: Data offset              = 20 bytes (4 bits)
  TCP: Reserved Bits:           - Reserved for Future Use (3 bits)
  TCP: ECN Nonce-Sum:           = 0 (1 bit)
  TCP: Flags                    = 10
  TCP:                0... .... = (Congestion Window Reduced (CWR) NOT set)
  TCP:                .0.. .... = (ECN Echo NOT Set)
  TCP:                ..0. .... = (No urgent pointer)
  TCP:                ...1 .... = Acknowledgment
  TCP:                .... 0... = (No push)
  TCP:                .... .0.. = (No reset)
  TCP:                .... ..0. = (No SYN)
  TCP:                .... ...0 = (No FIN)
  TCP: Window                   = 65535
  TCP: Checksum                 = 83D1 (should be 4786)
  TCP: Urgent pointer           = 0
  TCP: No TCP options
  TCP:
DLC:  Frame padding= 6 bytes
00000000: 00 48 41 44 44 52 00 11 25 18 0a db 08 00 45 00  .HADDR..%..Û..E.
00000010: 00 28 29 52 40 00 80 06 4d c7 c0 a8 01 02 c0 a8  .()R@...MÇÀ¨..À¨
00000020: 01 64 05 11 1f 98 f7 1d c8 cd 00 03 00 00 50 10  .d....÷.ÈÍ....P.
00000030: ff ff 83 d1 00 00 00 00 00 00 00 00              ÿÿ.Ñ........
```

```
TCP: ----- TCP header -----
TCP:
TCP: Source port                 =  8088
TCP: Destination port            =  1297
TCP: Sequence number             = 196608
TCP: Next expected Seq number= 196632
TCP: Acknowledgment number       = 4145924301
TCP: Data offset                 = 20 bytes (4 bits)
TCP: Reserved Bits:              - Reserved for Future Use (3 bits)
TCP: ECN Nonce-Sum:              = 0 (1 bit)
TCP: Flags                       = 10
TCP:                   0... .... = (Congestion Window Reduced (CWR) NOT set)
TCP:                   .0.. .... = (ECN Echo NOT Set)
TCP:                   ..0. .... = (No urgent pointer)
TCP:                   ...1 .... = Acknowledgment
TCP:                   .... 0... = (No push)
TCP:                   .... .0.. = (No reset)
TCP:                   .... ..0. = (No SYN)
TCP:                   .... ...0 = (No FIN)
TCP: Window                      = 65535
TCP: Checksum                    = 345D (correct)
TCP: Urgent pointer              = 0
TCP: No TCP options
TCP: [24 Bytes of data]
TCP:
DLC:  Frame padding= 1 bytes
00000000: 00 11 25 18 0a db 00 48 41 44 44 52 08 00 45 00  ..%..Û.HADDR..E.
00000010: 00 40 29 52 40 00 80 06 4d af c0 a8 01 64 c0 a8  .@)R@...M¯À¨.dÀ¨
00000020: 01 02 1f 98 05 11 00 03 00 00 f7 1d c8 cd 50 10  ..........÷.ÈÍP.
00000030: ff ff 34 5d 00 00 0d 0a 45 44 54 50 20 45 4e 43  ÿÿ4]....EDTP ENC
00000040: 32 38 4a 36 30 20 44 72 69 76 65 72 3e 00 41     28J60 Driver>.A
```

■ SCREEN SHOT 5. The next expected seq number is actually not part of the TCP protocol. The sniffer is assisting the user here by informing him or her that 24 bytes of data has been passed to the laptop from the Frame Thrower II. If you include the null (0x00) terminating character, the Telnet banner data is 24 bytes in length.

sequence (0D 0A).

At this point, the TCP application can step in. In the minimal TCP/IP stack code I have provided via the *Nuts & Volts* website (**www.nutsvolts.com**), the TCP application is a simple echo routine. Each character entered in the Telnet session at the laptop is received by the Frame Thrower II and echoed back to the laptop.

THE FRAME THROWER II HARDWARE

Much of what has to be said about the Frame Thrower II hardware is stated schematically by Schematic 1. The EDTP Frame Thrower II is an integrated 10Base-T Ethernet module. The Frame Thrower II is based on the Microchip ENC28J60 Ethernet Engine IC, which interfaces to just about any host microcontroller that supports a standard SPI interface. To allow for easy experimentation and application development, all of the PIC18F8722 microcontroller pins are brought out to standard .100-inch center holes.

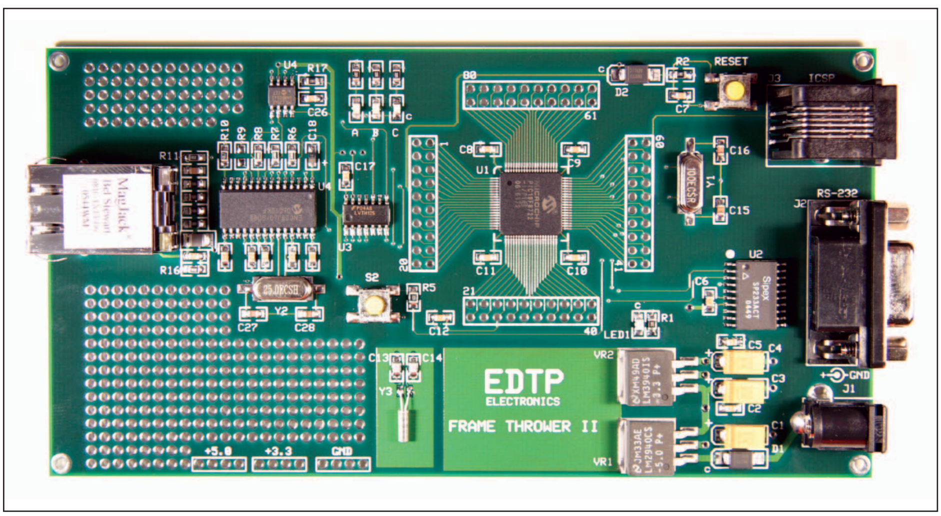
The Frame Thrower II comes equipped with three onboard status LEDs, which can be illuminated or extinguished under program control. A 10 MHz crystal is the seed for the 40 MHz PLL clock that drives the PIC18F8722 microcontroller. The ENC28J60 requires a 25 MHz crystal for its internal clock and, as you can see in Photo 1, the Frame Thrower II circuitry includes the 25 MHz clock source.

No communications project is complete without a fully functional serial port and a Sipex SP233 forms the heart of the Frame Thrower II's regulation RS232 interface. To keep things as simple as possible on the Ethernet side of the equation, the Ethernet physical interface is built around a consolidated magnetics/RJ-45 MagJack assembly from Bel Stewart.

To run at full speed, the PIC18F8722 must be powered from a +5.0 VDC supply. However, the ENC28J60 is strictly a +3.3 VDC part. To accommodate operation of both the PIC18F8722 and the ENC28J60, the Frame Thrower II incorporates a dual +5.0 VDC/+3.3 VDC regulated power supply subsystem.

If your application is time sensitive, an optional real-time clock can be implemented by adding a 32.768 kHz clock crystal and a couple of 20 pF capacitors to the Frame Thrower II printed circuit board. A 25LC256 EEPROM has been included to allow the network programmer to store web pages. However, the 25LC256 can be used for anything the network programmer (that's you, by the way) desires.

I would normally provide an ExpressPCB layout for the Frame



■ PHOTO 1. The EDTP Electronics, Inc., Frame Thrower II is a complete embedded Ethernet node. Not only does this little bugger run our simple TCP/IP code, it can also run the more sophisticated Microchip TCP/IP stack.

SOURCES

■ *EDTP Electronics, Inc.* — Frame Thrower II
www.edtp.com or
www.nerdvilla.com

Thrower II. However, the Frame Thrower II is built on a four-layer printed circuit board and the cost for you to purchase a single board is prohibitive. So, I've made arrangements to provide you with a kit of Frame Thrower II parts or assembled Frame Thrower II units by way of my website **www.nerdvilla.com** or the EDTP Electronics, Inc., website at **www.edtp.com**.

TIME TO SET THE FIN FLAG

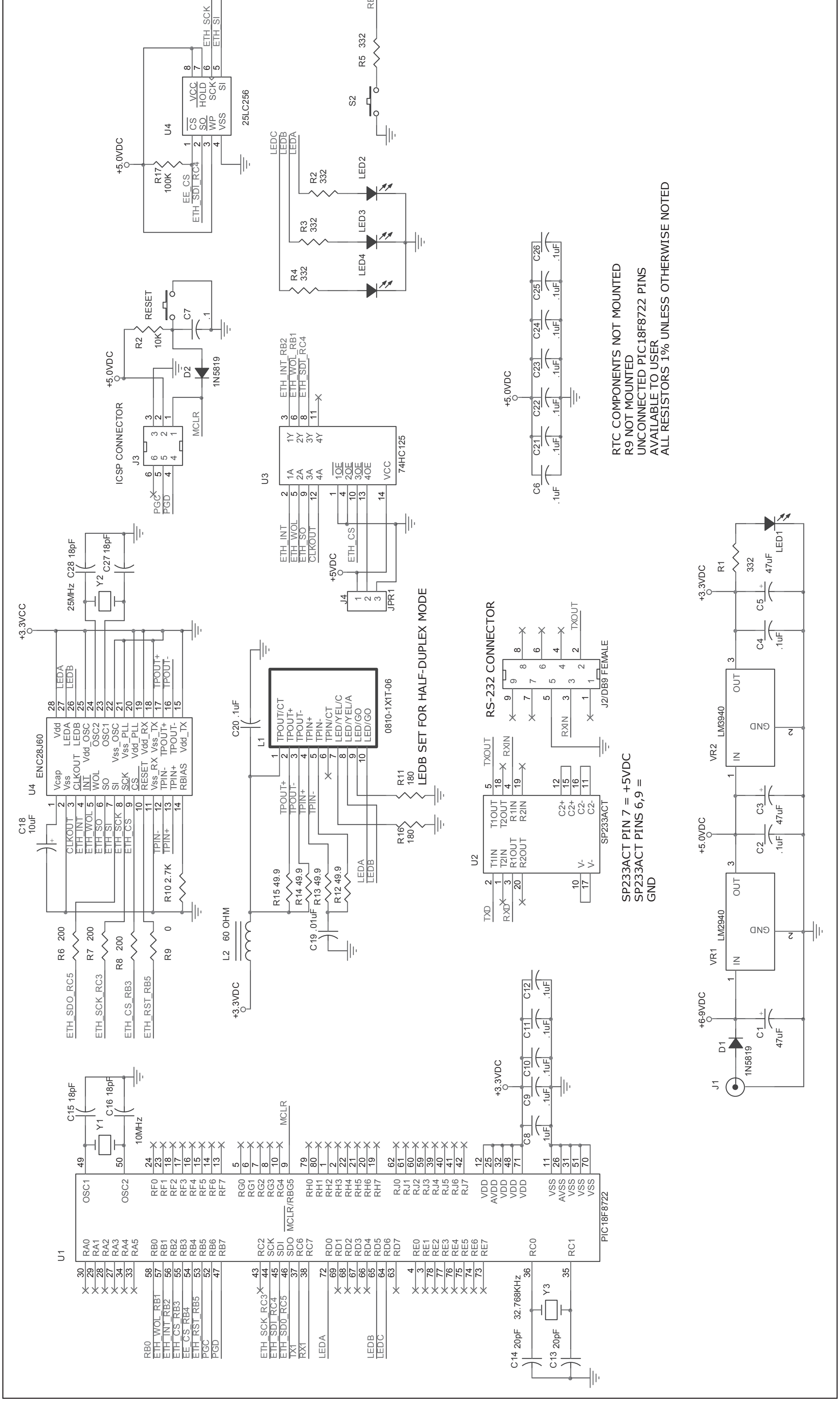
You now have two Internet protocols under your belt. If your application only requires the transmission and reception of simple data packets, UDP is the way to go. However, if you want to serve web pages or communicate via Telnet, you'll need to put TCP/IP into action. While we're on the subject of communicating, I have a new email address — peter@nerdvilla.com.

Now that you see how simple TCP/IP is under the covers, I'm sure you won't have any trouble incorporating TCP/IP into your Design Cycle. **NV**

ABOUT THE AUTHOR

■ Peter Best can be contacted via email at **peterbest@cfl.rr.com**

■ SCHEMATIC 1



■ BY CHUCK HELLEBUYCK

REAL TIME CLOCK

ONE OF THE MORE INTERESTING THINGS I've discovered about the readers of *Nuts & Volts* is their diversity. Through the emails I've received from writing this column, I've discovered that both hobbyists and professionals read these articles.

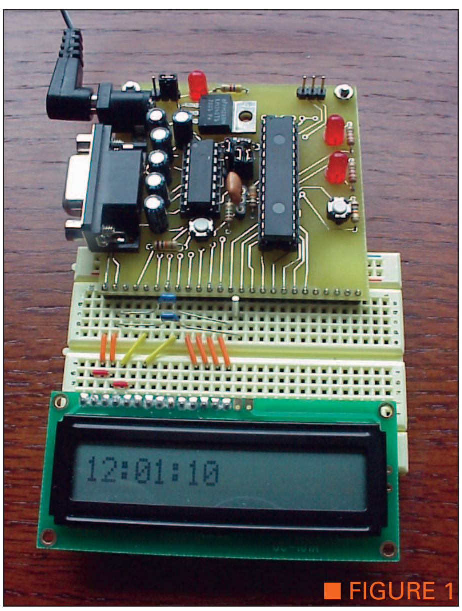
Readers vary from complete beginners to those with lots of experience programming PICs, but who still read to pick up some tips. It makes it hard to write an article to suit everybody, but I have learned that my method of writing snippets of code with a specific, simple purpose is appreciated by all.

I try to follow that method here and in all my books. This allows the beginner to see how to accomplish something and also allows the experienced PIC user to easily expand the code for their particular application.

With that in mind, I want to introduce using the PIC Timer1 — driven from its own clock crystal — to produce a simple clock (or as it's known in the microcontroller world, a "real time" clock.)

When you are talking about the bit speed or processor speed or even instruction cycle timing, it's often referencing the crystal or resonator running the microcontroller. It will be expressed in MIPS or Million Instructions per Second and it refers to the microcontroller's internal clock.

This isn't the time we humans use to determine if we can go home from work or if we should turn on the TV to catch the final game of the Stanley Cup (I'm a big hockey fan). Therefore, this human clock is considered the "real time" clock (RTC). There are numerous RTC chips available and some are RTCC or Real Time Clock and Calendar chips and, in the future, I'll talk about those as well, but in many applications, all you need is time and if you can do it easily with the PIC *already* in your application, why spend the extra board space and extra cost for a separate clock chip?



■ FIGURE 1

TIMER1

Inside many of the Microchip PICs exists a 16 bit timer called Timer1. One of the features of this timer is the ability to run it from a separate crystal than the rest of the microcontroller. This means it can actually run when the PIC is doing something else at a totally different speed. It can also be set up to run when the rest of the PIC is in low power sleep mode. This offers many advantages to your next PIC design because now you can add an accurate timebase even if you don't need a real time clock.

NOTE:
■ The complete software listing is available on the *Nuts & Volts* website at **www.nutsvolts.com**

The Timer1 is 16 bits wide so it can count from 0 to 65535. On the 65536th clock pulse, it rolls over (or overflows) to zero again and sets an interrupt bit (also referred to as a flag). In the PIC16F876A I like to use, the interrupt flag is the TMR1IF bit of the PIR1 register. By monitoring this bit, we can tell when Timer1 has overflowed.

The PIC offers the option of creating an interrupt when the timer overflows. This allows the software to perform a function automatically in the interrupt service routine every time an overflow occurs. To make this useful for a real time clock, we have to do a little calculation.

Crystals come in various frequencies, but clock crystals are mostly 32.768 kHz or 32,768 pulses per second. If we were to run the Timer1 from a 32.768 kHz crystal, then Timer1 would overflow to 65536 every two seconds (32,768 pulses per second times 2). For a real time clock, we would prefer it to overflow every second and we can make that happen by presetting Timer1 to $80 hex or %10000000 binary. This way, on the 32,769th pulse, the Timer1 overflows and the interrupt occurs.

INTERRUPTS IN ASSEMBLY

Because accuracy is important in a real time clock application, this is a great opportunity to introduce assembly language interrupts inserted into a Basic language program. The problem with higher level languages is the final assembly language code doesn't easily clarify how many internal clock cycles each high level instruction takes. This program needs to respond quickly to the Timer 1 overflow so, in cases where you need to react quickly and know exactly how many clock cycles occurs, assembly language is the answer. The Microchip PICs make this even easier because each assembly language command takes only one internal clock cycle (except for branches and GOTOs).

To keep this clock accurate, I will use a small section of assembly language inserted into a PICBasic Pro program to handle the Timer1 overflow and interrupt. This way, very few clock cycles are used to handle the overflow interrupt.

At this point, you might be thinking, I just learned how to program PICs in Basic, why rush into learning assembly? In my opinion, Basic better prepares you for assembly than any other method. Basic allows you to get a program working inside a PIC along with learning how the programmer works, how to wire the resonator or crystal, how to set up the configuration, and how to establish register and RAM (variables). It even forces you to read the data sheets a little bit. Writing assembly code after all that is just learning a few new commands that work on the bit level just like the HIGH and LOW Basic commands do.

PROJECT DETAILS

For this project, I'm going to drive a 2 x 16 LCD module to display the time generated by the Timer1 overflow. The software of the main loop will handle converting the one-second overflow into a meaningful clock display of hours : minutes : seconds. The hardware involved is not much different than my original article that started this series back in January. We do most of it in software.

The completed project is shown in Figure 1. I built it on a breadboard again, using one of my original prototype Ultimate OEM modules, but a standalone PIC is a simple alternative. The schematic in Figure 2 shows my setup. The key component is the 32.768 kHz crystal between the C0 and C1 pins of the PIC. It also has two 22 pf capacitors from the crystal leads to ground. The rest of the circuit drives a 2 x 16 LCD. (I've covered driving before.)

SOFTWARE LISTING

The software is written for PICBasic Pro. I wanted to show how to mix assembly language and BASIC to achieve an accurate real time clock. To get a quick response to a Timer1 overflow, I chose to use assembly language interrupts. This may seem a bit complex if you don't know assembly language, but it's really not that difficult to understand. Let's start at the beginning to explain this software.

I start by defining the fact I'm using a bootloader to program the PIC and I'll run that PIC at 20 MHz. Later on we'll set up the 32.768 kHz crystal running Timer1.

```
DEFINE LOADER_USED 1      ' Needed for bootloader redirect
DEFINE OSC 20             ' 20 MHz resonator
```

Next, all the DEFINES for the LCD are established. I'm using PortB bits 4-7 for the data communication, bit 3 for the E line, and bit 0 for the RS line. The R/W pin is just grounded on the LCD for "write-only" operation. One of the key setups here is the DEFINE LCD_DATAUS 100. I found the default of 50 microseconds did not work well with my LCD. I had to slow down the data transmission. This is why I like to define all of the LCD setup rather than rely on the defaults.

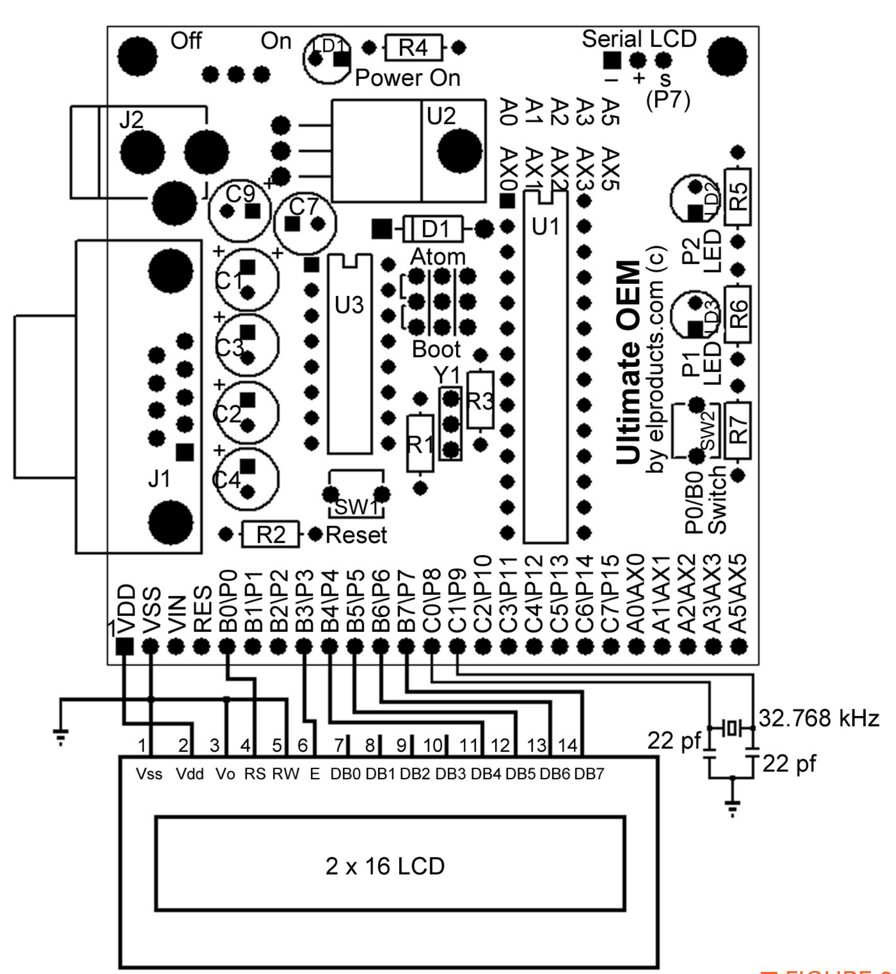


FIGURE 2

```
DEFINE LCD_DREG  PORTB      ' Define PIC port used for LCD
                            ' Data lines
DEFINE LCD_DBIT  4          ' Define first pin of portb
                            ' connected to LCD DB4
DEFINE LCD_RSREG PORTB      ' Define PIC port used for RS
                            ' line of LCD
DEFINE LCD_RSBIT 0          ' Define Portb pin used for RS
                            ' connection
DEFINE LCD_EREG  PORTB      ' Define PIC prot used for E
                            ' line of LCD
DEFINE LCD_EBIT  3          ' Define PortB pin used for E
                            ' connection
DEFINE LCD_BITS  4          ' Define the 4 bit
                            ' communication
                            ' mode to LCD
DEFINE LCD_LINES 2          ' Define using a 2 line LCD
DEFINE LCD_COMMANDUS 2000   ' Define delay between sending
                            ' LCD commands
DEFINE LCD_DATAUS 100       ' Define delay time between
                            ' data sent.
```

Now we get to the more difficult part to understand. We have to define the interrupt handler for assembly with the command below. This is different from the Basic language interrupt handler setup. The label "myint" in this case is the label in the assembly code section of this program.

```
' Define Assembly Language interrupt handler
DEFINE   INTHAND myint    ' Define interrupt handler
```

Because we may be interrupting some assembly code that is actually compiled Basic code, we need to save the special registers that control the PIC internal operation. Even if you write in pure assembly, you should do this. PICBasic Pro already establishes some reserved names for doing this. In the PICBasic Pro manual, these special function register names are listed. Below the program, a storage area for saving this information is established.

```
' Establish special variables for assembly interrupts
wsave    VAR   BYTE $20 system   ' W storage in bank0
wsave1   VAR   BYTE $a0 system   ' W storage in bank1
wsave2   VAR   BYTE $120 system  ' W storage in bank2
wsave3   VAR   BYTE $1a0 system  ' W storage in bank3
ssave    VAR   BYTE bank0 system ' Status register save
psave    VAR   BYTE bank0 system ' PCLATH register save
```

For our program, we will count how many times Timer1 overflows and store that count in the variable "TICK." Because the PIC16F876A is arranged in several banks of 2K, we want everything to be in the first bank (Bank 0), so we don't have to play with the Status register which selects which bank to work in. We do that with the "bank0" directive after establishing "TICK" as a byte variable. I wanted to be able to easily adjust the overflow of the Timer1 so I also created a variable "Preload" to store the value I preset Timer1 to.

```
TICK    VAR    BYTE bank0 ' make sure that the variables
                          ' are in bank 0 if they are to
                          ' be used in the interrupt
                          ' handler
PRELOAD var byte bank0    ' value for timer preload
```

The variables for seconds, minutes, and hour are established as bytes but we will only be using these in the Basic section of code, so we don't need to set them to a special bank.

```
seconds    VAR BYTE            ' Elapsed seconds
minutes    VAR byte            ' Elapsed minutes
hour       VAR byte            ' Elapsed hours
```

I didn't add switches to the design so I could set the clock. I'll leave that for the reader to add. Instead, I just preset the starting time to 12:00:00 by presetting the seconds, minutes, and hour variables. I also preset "Preload" to $80 hex.

```
minutes = 0            ' Preset time to 12:00:00
seconds = 0
hour = 12
preload = $80          ' Preset timer to 80 hex
```

To properly set up Timer1 to run from an external 32.768 kHz crystal, I have to preset the T1CON register inside the PIC. Each bit has a specific function and, if you look in the PIC16F876A data sheet, you'll see that %00001111 in that eight-bit register turns the timer on, sets the prescaler to 1:1, and establishes the clock source as external. That is all it really takes to make Timer1 run from a 32.768 kHz crystal.

```
T1CON = $0F            ' Turn on Timer1, prescaler=1,
                       ' External 32.768 kHz
```

We also want to use assembly language interrupts, so the INTCON and PIE1 registers have to be set up to turn on the Timer1 overflow interrupt. Those registers are set up below.

```
INTCON = $C0      ' Enable global interrupts, peripheral
                  ' interrupts
PIE1 = $01        ' Enable TMR1 overflow interrupt
```

Next, the assembly code and interrupt routine need to be placed at the top of the code so they end up in bank zero of memory. Therefore, we have to add a jump or GOTO command to leap over the assembly code interrupt when the PIC is running through here for the first time. Remember, we only want to run assembly and the interrupt routine when the Timer1 overflows.

```
GoTo init         ' jump over the interrupt handler
                  ' and subroutine
```

Ah, the dreaded assembly code! We start by telling the PICBasic Pro compiler assembly is coming with the "ASM" command. Notice how the comment following "ASM" switches to a ";" from a " ' " as a lead. This is because the assembler doesn't recognize " ' " as a comment initiator. Assembly code goes right past the Basic Compiler and directly into the assembly file created by the Basic

Compiler. This is why we have to follow the rules of the assembler.

```
' *******************
' Assembly language interrupt handler
' *******************

Asm              ; Tell PicBasic Pro Assembly
                 ; Commands follow
```

The label "myint" which matches our earlier DEFINE is established, followed by a bunch of commented out lines. I got these direct from the PICBasic Pro manual and for PICs with 2K of space or less, these need to be used. This is where the register data is stored before running the interrupt routine. Because the PIC16F876A is more than 2K, I can leave them commented out. PICBasic Pro takes care of this storage for parts greater than 2K.

```
myint

; Uncomment the following if the device
; has less than 2K of code space
     ;movwf wsave         ; Save W
     ;swapf STATUS, W     ; Swap STATUS to W
                          ; (swap avoids changing STATUS)
     ;clrf STATUS         ; Clear STATUS
     ;movwf ssave         ; Save swapped STATUS
     ;movf PCLATH, W      ; Move PCLATH to W
     ;movwf psave         ; Save PCLATH
```

The Timer1 overflow occurred or the program would not be here. Therefore, the program needs to increment the TICK variable and preload the Timer1 to $80 hex again. That is all that happens here. I commented out the line where I could preload the Timer1 with $80 hex directly and use the variable Preload instead. I did this to show how the Basic code and assembly code can share data.

The Timer1 is a 16 bit register controlled by the TMR1H and TMR1L byte registers. Since we only need to update the high byte, we load the $80 not $8000 into the TMR1H register with a series of assembly commands. We first load the Working (W) register with $80 (080H) and then move the contents of the W register to the TMR1H register.

```
; Set the high register of Timer1 to cause an interrupt
; every 32768 counts (65536-32768 = 32768 or $8000).

;      movlw     080h     ; Prepare to set TMR1
                          ; high register
       movf   _PRELOAD,W  ; Preload Timer1 for
                          ; 1 second overflow
       movwf TMR1H        ; Set TMR1H to 80h
```

The Tick variable needs to be incremented. We do that with the INC assembly command acting on the _TICK variable. Notice how TICK and Preload both have an underscore in front of it. This is how a variable established in Basic is accessed in assembly.

```
     incf   _TICK,F       ; Increment TICK count
                          ; (number of overflows)
```

We have to clear the interrupt flag so the program doesn't jump back to the interrupt routine again. We do this with a bit clear file command (BCF) acting on the PIR1 register where the Timer1 interrupt flag is located.

```
     bcf      PIR1, 0    ; Clear interrupt flag
```

Even though PICBasic Pro stored the important register data for us, we still have to add code to put it back to where it was before we interrupted. Here is the section that does that.

```
     movf     psave, W  ; restore the state
                        ; of everything
     movwf    PCLATH
     swapf    ssave, W
     movwf    STATUS
     swapf    wsave, F
     swapf    wsave, W
     retfie             ; Return from interrupt
```

The dreaded assembly code is done and we indicate that with an ENDASM command.

```
EndAsm                    ' End of assembly code
```

Now the subroutine is written to update the clock data based on the number of counts in the TICK variable. It starts at the label "get_time." The first act is to disable interrupts during this update mode. We do that by clearing the PIE1 bit. Do you see how similar this is to BCF PIR1, 0 at the end of the assembly section? Basic and assembly have similarities.

```
' **********************
' PicBasic subroutine to update the minutes
' and seconds variables
' **********************
get_time:
' Read the timer1 overflows as TICK and then add
' to the seconds variable. Increment Minutes and Hours
' as necessary.

     PIE1 = 0    ' Mask the interrupt while
                 ' we're messing with TICK
```

Now we go though a series of math functions to take the TICKs variable and convert it to seconds, minutes, and hours. The seconds is an easy addition. After we are done with updating seconds, we can turn the interrupt back on so we don't miss any. The minutes and hours will be updated based on the seconds variable, anyway.

```
     seconds = seconds + tick    ' Add the accumulated
                                 ' seconds
     tick = 0                    ' Retain the left-over
                                 ' ticks
     PIE1 = $01                  ' Interrupt on again
```

The minutes is equal to its old value plus the new number of seconds divided by 60. Anything left over in that

division is the odd number of seconds less than one minute, but we can find that value separately by dividing and looking for the remainder only with the "//" math function. We do this same method again to get the hours and minutes aligned properly.

```
' Add the accumulated minutes
     minutes = minutes + (seconds / 60)

' Retain the left-over seconds
     seconds = seconds // 60

' Add the accumulated hours
     hour = hour + (minutes/60)

' Retain the left -over minutes
     minutes = minutes//60
```

I wanted 12 hour time, not 24 hour, so I added an IF-Then statement to make that happen.

```
     if hour > 12 then
     hour = hour - 12
     endif
```

Because this was a subroutine, we need to add the RETURN command.

```
Return                    ' Return to the main program
```


Finally, after all that, we get to the main loop of code and it's very short. We start with the "init:" label and just clear the LCD.

```
init:
' *****************************************************
' Begin program code here. The time will be updated
' when you call the get_time routine.
' *****************************************************

LCDOut $fe,1             ' Clear LCD
```

The main loop of code is at the "main" label. The program jumps to the "get_time" subroutine to get the latest seconds, minutes, and hour values and then displays them on the LCD.

```
main:

     GoSub get_time      ' Update minutes and seconds

'**** Display the elapsed time ******
     LCDOut $fe, 2, DEC2 hour, ":", DEC2 minutes, ":",
DEC2 seconds
```

The program loops back to the main label and does it over and over again.

```
     GoTo main           ' Repeat main loop

     End
```

That's all there is to making a real time clock with PICBasic Pro and some assembly thrown in for accuracy.

NEXT STEPS

A great place to improve this program is to add switches for presetting the seconds, minutes, and hour variables. This way, the clock can be set to any time you desire. Another great action would be to add some kind of compare routine to make an alarm clock. When the hour and minutes variables match a preset value, a buzzer can be activated or a relay driven. There are a lot of options you can choose from once you have the basic clock working.

SUMMARY

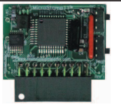
I hope the addition of a little assembly wasn't too much for the beginners out there. I wanted to show that assembly now and then is very helpful. Learning PIC assembly is a good thing and you will find it's not that tough to understand once you've written a few successful Basic programs for the PIC.

Let me know how you liked this article and what else you'd like to see in future articles. I'm getting a lot of feedback that helps me determine what readers want. If you don't tell me, I won't know what you want to see. Email me at chuck@elproducts.com. **NV**

CLASSIFIEDS

MICRO CONTROLLERS

MicroStamp11
World's Smallest 68HC11 Microcontroller Module!



- telemetry
- microrobotics
- smart toys
- animatronics
- model railroading
- home automation

- tiny (1 by 1.4 in.), light-weight (0.5 oz.)
- on-board 5V reg., crystal, & reset chip
- choice of 8K or 32K EEPROM
- or 32K RAM + 32K EEPROM (64K version)
- SCI, SPI, Output Compare and Input Capture channels, timer, pulse accumulator
- all 14 I/O lines and 2 interrupt lines brought out to versatile 20-pin connector
- program in BASIC, assembler, or C
- easy code-loading with Docking Module
- Starter Packages:*
  - 8K EEPROM (#MS11SP8K).......$49
  - 32K EEPROM (#MS11SP32K)...$77
  - 32K EE/32K RAM (MS11SP64K)$90

** includes MicroStamp11, manual, PC software (assemblers, SBASIC compiler, MicroLoad utility, and sample programs), serial cable, Docking Module, & accessories.*

www.technologicalarts.com
Toll-free (USA & Canada): 1-877-963-8996
Visa • MasterCard • Discover • Amex

PIC® Microcontroller Books
4 Titles
Beginner to Advanced
Table Of Contents
Ordering Info
On Web Site
http://www.sq-1.com
SQUARE 1 ELECTRONICS
(208) 664-4115
Hayden, ID USA

Call us TODAY to start your subscription to *SERVO Magazine!*
877.525.2539

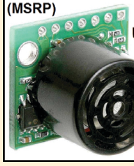
SATELLITE

FREE Satellite TV Buyer's Guide

Home of DishSkinz
C-band • Ku-band
4DTV • MPEG-2
Get it all with just one call!
800-500-9275
Skyvision®
www.skyvision.com

ROBOTICS

$29.95 MaxSonar-EZ1
(MSRP)



High Performance Ultrasonic Range Finder
- serial, analog voltage & pulse width outputs
- lowest power - 2mA
- narrow beam
- very easy to use!

www.maxbotix.com

ROBOTIC CONTROLLERS
Robot Bases. PIC, OOPic, Stamp and servo controllers. New Walking Machine Controller. Chips, kits, assembled.
www.oricomtech.com

SERVO
The Future of Robotics is at Your Fingertips!
Subscribe Today!
www.servomagazine.com

CONNECTORS/WIRE/CABLE

The RF Connection
213 N. Frederick Ave., Ste. 11NV
Gaithersburg, MD USA 20877
http://www.therfc.com/

Complete Selection of MIL-Spec Coax, RF Connectors and Relays

UG-21B/U N Male for RG-213/214 .$5.00
UG-21D/U N Male for RG-213/214 .$3.25
N Connectors for 9913/Flexi4XL/9096
UG-21B/9913 .....$6.00 / Pins Only......$1.50
UG-21D/9913 ..$4.00 / Extra Gasket.$0.75
Amphenol 83-1SP-1050 PL-259 $0.90
UG-176/U Reducer RG-59/8X, $0.25 or 5/$1.00
UG-175/U Reducer RG-58/58A, $0.25 or 5/$1.00
Silver Teflon PL-259/Gold Pin, $1.00 or 10/$9.00

MIL-Spec Coax Available (Teflon, PVC IIA)
New Product: Belden 9913F, 9913 with High Density PE Foam dielectric, stranded center cond. and Duobond III Jacket $0.80/ft or $76.00/100ft
Also New: 9092, RG8X with Type II Jacket. Intro Price $23.00/100ft

Call for Specials of the Month
Full Line of Audio Connectors for Icom, Kenwood, and Yaesu

8 Pin Mike Female	$2.50
8 Pin Mike Male Panel	$2.50
13 Pin DIN for Kenwood	$2.75
8 Pin DIN for Icom	$1.00
8 Pin DIN for Kenwood	$1.50

Prices Do Not Include Shipping

Orders	**800-783-2666**
Info	**301-840-5477**
FAX	**301-869-3680**

SCSI
I-U320 50Pin-68Pin-80Pin
1 to 8 Bay Case Enclosures
Adapters Cables Terminators
Low Prices - Qty Discounts!
(Also FireWire, USB, Video)
www.mcpb.com

www.servomagazine.com

MISC. FOR SALE

PLANT CLOSED
MUST SELL: Megohmmeter, Chart Recorders, Bridges, 10-75KVA Transformers, Spot Welders, Vacuum Pumps, Power Supplies, Leak Detectors, Test Equip, Industrial Ovens, Sputterer, Breaker Boxes to 600Amp, Constant Voltage Trans, Polariscope, Glass Working Equip, Packing Equip
Details: www.isesurplus.com or call 956-351-5550 / 888-350-5555 (toll free)

PUBLICATIONS

DETAILED TECHNICAL MANUALS
- Current Activated Switch Project ... $10
- Surge Protector Project ... $10
- Car Battery Float Charge Project ... $10
- Alternator Voltage Regulator Project ... $15
- Common Electronic Components ... $15
- Linear Power Supply Design ... $15

USPS postage included (USA only)
mlbunyard@door.net
Mike Bunyard, PO Box 6454, Lubbock, TX 79464-45

CNC Book
Easy CNC
G-code
Bit map draw programs
Bit map image converter
Bit map to G-code
CAD - machine drawings
CAM - DXF to G-code
http://www.cncintro.com
SQUARE 1 ELECTRONICS
(208) 664-4115
Hayden, ID USA

MILITARY SURPLUS

Electronic Military Surplus

Fair Radio Sales
Website: fairradio.com
E-mail: fairradio@fairradio.com
Phone: 419-227-6573
Fax: 419-227-1313
2395 St Johns Rd
Lima, OH 45804
Visa, Mastercard, Discover
Address Dept. N/V

30 Foot Mast Kit, $139.50
PRC-47 HF Transceiver, $150.00
6 Position Ant Tap Switch, $3.95
88MFD 1.2KV Capacitors, $17.50
URM-25 HF Signal Generator, $110
RT-834 Ground Radio 2-30MHz, $300
Yagi Antenna 928-960MHz, $19.95
ME-165 Power Meter 600W, $125.00
PRC-25 Backpack Radio, $300.00
HP-331A Distortion Analyzer, $225
5KV Doorknobs 10, 25, 50, 75pf, $6 ea
Various Velleman Electronic Kits

***Many** more items available*
See website for more info

Send for our Latest Catalog !!

Radios - Test Equipment - Tubes - Antennas

COMPUTER HARDWARE WANTED

DEC EQUIPMENT WANTED!!!
Digital Equipment Corp. and compatibles.
Buy - Sell - Trade
CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

COMPONENTS

AMATEUR RADIO

BUSINESS SERVICES

DESIGN/ENGINEERING SERVICES

LCDs/DISPLAYS

AUDIO/VIDEO

 Continued from page 6

proved problematic. It was then that I discovered the handy Texas Instruments (formerly Power Trends) PT5101 integrated switching regulator. This little module (about one inch square) is a virtual drop-in replacement for the 7805 in most circuits, but is far more efficient (90+ percent) due to its switch-mode (versus linear) operation. It can deliver +5 volts at up to 1 ampere of output current with input voltages ranging anywhere from 9 to 38 volts, and all this with no heatsink! The only external component required is a 100 µF capacitor from the output pin to ground, which is a common requirement for many regulators, anyway. It is available in vertical (N suffix) and horizontal (A suffix) through-hole versions, as well as surface mount (C suffix). The only downside? It costs about $12 from Digi-Key. But for many of your readers' applications, it might be just the ticket! It is now my first choice when I need on-board voltage regulation. By the way, there are other output voltages available too, such as +3.3, +9, +12, and -12 (yes, inverting), as well as boost versions (the PT504x series) that can generate your other on-board voltage needs with a +5 volt supply as input! These devices are a great alternative to a heatsink in many designs. Thank you for the opportunity to share my experience with your readers.

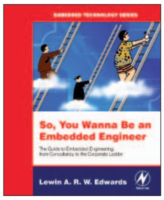
Brent Hauser, Eagan, MN

Selected Titles for the Electronics Hobbyist and Technician

The Nuts & Volts Hobbyist BOOKSTORE

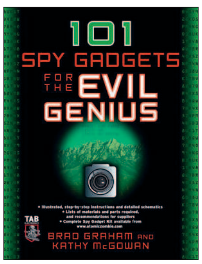
ELECTRONICS

So You Wanna Be an Embedded Engineer
by Lewin Edwards



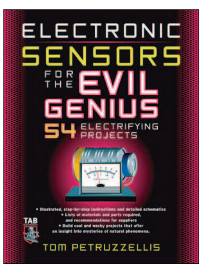
In this new, highly practical guide, expert embedded designer and manager Lewin Edwards answers the question, "How do I become an embedded engineer?" Embedded professionals agree that there is a treacherous gap between graduating from school and becoming an effective engineer in the workplace, and that there are few resources available for newbies to turn to when in need of advice and direction. This book provides that much-needed guidance for engineers fresh out of school, and for the thousands of experienced engineers now migrating into the popular embedded arena. **$39.95**

101 Spy Gadgets for the Evil Genius
by Brad Graham/Kathy McGowan



Utilizing inexpensive, easily-obtainable components, you can build the same information gathering, covert sleuthing devices used by your favorite film secret agent. Projects range from simple to sophisticated and come complete with a list of required parts and tools, numerous illustrations, and step-by-step assembly instructions. **$24.95**

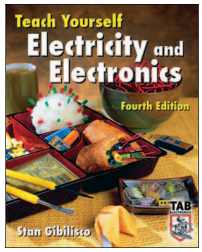
Electronic Sensors for the Evil Genius — 54 Electrifying Projects
by Thomas Petruzzellis



Nature meets the Evil Genius via 54 fun, safe, and inexpensive projects that allow you to explore the fascinating and often mysterious world of natural phenomena using your own home-built sensors. Each project includes a list of materials, sources for parts, schematics, and lots of clear, well-illustrated instructions. Projects include rain detector, air pressure sensor, cloud chamber, lightning detector, electronic gas sniffer, seismograph, radiation detector, and much more. **$24.95**

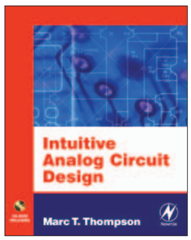
If you don't see what you need here, check out our online store at www.nutsvolts.com for a complete listing of the titles available.

Teach Yourself Electricity and Electronics — Fourth Edition
by Stan Gibilisco



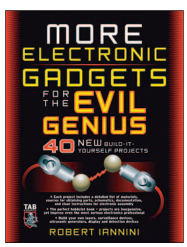
Learn the hows and whys behind basic electricity, electronics, and communications without formal training. The best combination self-teaching guide, home reference, and classroom text on electricity and electronics has been updated to deliver the latest advances. Great for preparing for amateur and commercial licensing exams, this guide has been prized by thousands of students and professionals for its uniquely thorough coverage ranging from DC and AC concepts to semiconductors and integrated circuits. **$34.95**

Intuitive Analog Circuit Design
by Marc Thompson



This book introduces analog circuit design with a minimum of mathematics. It gives readers an intuitive "feel" for analog circuit operation and rules-of-thumb for their design. The author uses numerous analogies from digital design to help readers whose main background is in digital make the transition to analog design. The application of some simple rules-of-thumb and design techniques is the first step in developing an intuitive understanding of the behavior of complex electrical systems. This book outlines some ways of thinking about analog circuits and systems that hopefully develops such "circuit intuition" and a "feel" for what a good, working analog circuit design should be. **$59.99**

MORE Electronic Gadgets for the Evil Genius
by Robert E. Iannini



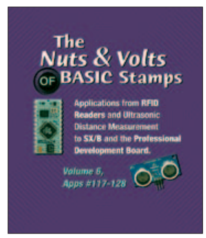
This much anticipated follow-up to the wildly popular cultclassic *Electronic Gadgets for the Evil Genius* gives basement experimenters 40 all-new projects to tinker with. Following the tried-and-true Evil Genius Series format, each project includes a detailed list of materials, sources for parts, schematics, documentation, and lots of clear, well-illustrated instructions for easy assembly. Readers will also get a quick briefing on mathematical theory and a simple explanation of operation along with enjoyable descriptions of key electronics topics. **$24.95**

MICROCONTROLLERS

Programmable Logic Controllers — Fourth Edition
by W. Bolton

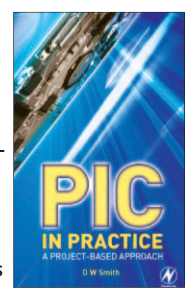
Now in its fourth edition, this best-selling text has been expanded with increased coverage of industrial systems and PLCs and more consideration has been given to IEC 1131-3 and all the programming methods in the standard. The new edition brings the book fully up to date with the current developments in PLCs, describing new and important applications such as PLC use in communications (e.g., Ethernet — an extremely popular system), and safety — in particular proprietary emergency stop relays (now appearing in practically every PLC based system). **$34.95**

Nuts & Volts of BASIC Stamps — Volume #6
by Jon Williams



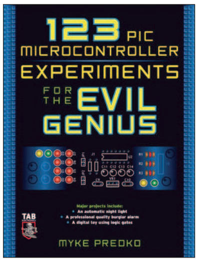
Nuts & Volts of BASIC Stamps — Volume 6 includes articles #117-128, written for 2005. Article topics consist of RFID Readers and Ultrasonic Measurement, SX/B and the Professional Development Board, the advanced MIDI receiver, programming the SX microcontroller in BASIC, mastering the MC14489 display driver, and more! The Nuts & Volts of BASIC Stamps books are a favorite Parallax technical pick and are a tremendous technical resource for all PBASIC programming projects. **$14.95**

PIC in Practice
A Project-based Approach
Second Edition
by David W. Smith



PIC in Practice is a graded course based around the practical use of the PIC microcontroller through project work. Principles are introduced gradually, through hands-on experience, enabling students to develop their understanding at their own pace. The book can be used at a variety of levels and the carefully graded projects make it ideal for colleges, schools, and universities. Newcomers to the PIC will find it a painless introduction, whilst electronics hobbyists will enjoy the practical nature of this first course in microcontrollers. **$29.95**

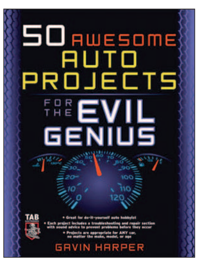
123 PIC Microcontroller Experiments for the Evil Genius
by Myke Predko



Microchip continually updates its product line with more capable and lower cost products. They also provide excellent development tools. *123 PIC Microcontroller Experiments for the Evil Genius* uses the best parts, and does not become dependent on one tool type or version, to accommodate the widest audience possible. Building on the success of *123 Robotics Experiments for the Evil Genius*, as well as the unbelievable sales history of *Programming and Customizing the PIC Microcontroller*, this book will combine the format of the evil genius title with the following of the microcontroller audience for a sure-fire hit. **$24.95**

AUTOMOTIVE

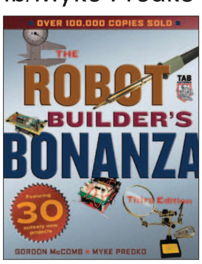
50 Awesome Auto Projects for the Evil Genius
by Gavin D J Harper



The Evil Genius format is the perfect "vehicle" for 50 incredible automotive projects that are compatible with any car, no matter what make, model, or year. Focusing on low-cost, easily obtained components, *50 Awesome Auto Projects for the Evil Genius* lists the items needed to complete each project along with a troubleshooting and repair section. **$24.95**

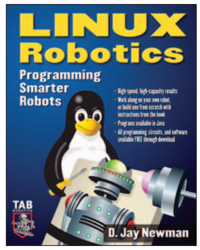
ROBOTICS

Robot Builder's Bonanza Third Edition
by Gordon McComb/Myke Predko



Everybody's favorite amateur robotics book is bolder and better than ever — and now features the field's "grand master" Myke Predko as the new author! Author duo McComb and Predko bring their expertise to this fully-illustrated robotics "bible" to enhance the already incomparable content on how to build — and have a universe of fun — with robots. Projects vary in complexity so everyone from novices to advanced hobbyists will find something of interest. Among the many editions, this book features 30 completely new projects! **$27.95**

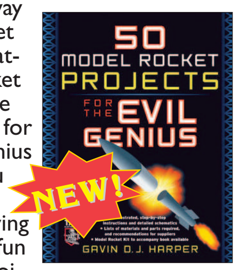
Linux Robotics
by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need, plus complete training in doing original programming. **$34.95**

PROJECTS

50 Model Rocket Projects for the Evil Genius
by Gavin D J Harper



The fun, hands-on way to learn about rocket science. Yes, as a matter of fact, is IS rocket science! And because this book is written for the popular Evil Genius format, it means you can learn about this fascinating and growing hobby while having fun creating 50 great projects. You will find a detailed list of materials, sources for parts, schematics, and lots of clear, well-illustrated instructions. **$24.95**

HOME COMPUTING

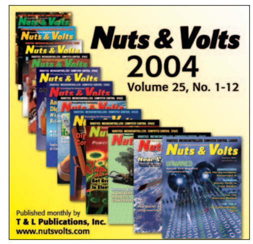
PC Upgrading and Troubleshooting QuickSteps
by Kirk Steers

This low-priced, fast reference uses color screenshots and brief instructions to show and explain how to fix all kinds of PC problems. Each chapter's "How to" list and color coded tabs make it easy to flip straight to specific tasks, such as routine maintenance, start-up issues, drive, monitor, and peripheral problems, Internet connections, upgrading, and more. Useful tips, reminders, shortcuts, and cautions are displayed in the margins so they don't break the flow of the book. **$16.99**

SERVO CD-Rom

Starting with the first *SERVO* issue — November 2003 — all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 1, issues 11-12 and Volume 2, issues 1-12, for a total of 14 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

Nuts & Volts CD-Rom



Here's some good news for *Nuts & Volts* readers! Starting with the January 2004 issue of *Nuts & Volts*, all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 25, issues 1-12, for a total of 12 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

AUDIO

Self on Audio — Second Edition
by Douglas Self

Whether you are a dedicated audiophile who wants to gain a more complete understanding of the design issues behind a truly great amp, or a professional electronic designer seeking to learn more about the art of amplifier design, there can be no better place to start than with the 35 classic magazine articles collected together in this book. Douglas Self offers a tried and tested method for designing audio amplifiers in a way that improves performance at every point in the circuit where distortion can creep in – without significantly increasing cost. **$44.95**

WE ACCEPT VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change. Ask about our 10% subscriber discount on selected titles.

TECH FORUM

This is a READER-TO-READER Column.

All questions *AND* answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving problems of a technical nature. Questions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and NO GUARANTEES WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals. Always use common sense and good judgement!

All questions and answers should be sent by email to forum@nutsvolts.com All *diagrams should be computer generated and sent with your submission as an attachment.*

QUESTIONS

To be considered, all questions should relate to one or more of the following:

1. Circuit Design
2. Electronic Theory
3. Problem Solving
4. Other Similar Topics

■ Be brief but include all pertinent information. If no one knows what you're asking, you won't get any response (and we probably won't print it either).

■ Include your Name, Address, Phone Number, and email. Only your Name, City, and State will be published with the question, but we may need to contact you.

■ No questions will be accepted that offer equipment for sale or equipment wanted to buy.

■ Selected questions will be printed one time on a space available basis.

■ Questions are subject to editing.

ANSWERS

■ Include in the subject line of your email, the question number that appears directly below the question you are responding to.

■ Payment of $25.00 will be sent if your answer is printed. Be sure to include your mailing address or we cannot send payment.

■ Only your Name, City, and State, will be printed, unless you say otherwise. If you want your email address included, indicate to that effect.

■ Comments regarding answers printed in this column may be printed in the Reader Feedback section if space allows.

>>> QUESTIONS

I have heard that florescent lights can be dimmed. The circuit is a constant voltage generator (120V) and a current controller that changes the frequency to lower the light intensity.

Can someone supply such a circuit that uses a variable frequency to change the amount of current flowing?

The original ballast is still being used in the circuit.

#9061 **Edwin Smith**
via email

I want to use an accelerometer as a motion sensor for a vehicle security system. I just want to measure the motion of the car.

Can someone please advise me on which accelerometer type (capacitive, temperature, etc.) is good for this application?

What are the advantages and disadvantages?

#9062 **Pono Lenyora**
via email

I have a question about heat dissipation vs. duty cycle. Power increases proportional to current squared, and linearly with respect to resistance. As duty cycle varies, should I use the square root of the on (off?) time to calculate power? (Is this not analogous to pulsed laser calculations where energy is related to pulse duration and can get quite high for short on times?)

Here is an interesting tidbit for all *Nuts & Volts* readers: In copper wire, the rated current is 0.003 times the circular mils. I saw this tidbit many years ago and was too dense to grab its significance but have since "got it" and am wondering how many techs know it and use it.

#9063 **Robert C. Gibson**
Aurora, IL

>>>> ANSWERS

[#7064 - July 2006]

I purchased a GOLDSTAR DC power supply model GP-305. There are some LEDs, switches, and terminals that I have not been able to figure out. Can anyone tell me the function of these or have a manual I might be able to copy?

The mystery elements are:

***1.** A switch marked EXT/INT.*
***2.** An LED marked C.C.*
***3.** An LED marked C.V.*
***4.** A external terminal strip with the following contacts /C1/C2/+/EXT/-/PL/GND.*

I have figured out GND.

The answers to Mr. Lang's questions are:

(1) This is for use with multiple units and determines which one is the control unit.

(2) An LED marked C.C. When on,

this indicates the unit is in "Constant Current" mode.

(3) An LED Marked C.V. When on, this indicates the unit is in "Constant Voltage" mode.

(4) An external terminal strip: "C1"- is connected to C2, when used multi-external VR of voltage control. "C2" is connected to C1, when used multi-external VR of voltage control. EXT "+" is connected to the outside VR of voltage control. EXT "-" is connected to the outside of voltage control. PL is the connection terminal for parallel operation.

All of the above is directly from the instruction manual for the GP-303/305/503/505 power supplies.

If Mr. Lang would like a copy of the manual he can email me with a fax number or mailing address. There would be no charges.

Merrill Hendrickson
wa2cll@aol.com

[#8064 - August 2006]

I want to build a 24 VDC power supply using the LM723 voltage regulator, at around 20-30 amps.

I can custom wind a large transformer for this project, but I need a schematic for the voltage regulation section. I would also like to know if I could use a power MOSFET instead of a bipolar transistor like the 2N3055. I believe that using a MOSFET with a low on resistance should lower the heat generated. Does anyone have an idea or schematics to do this?

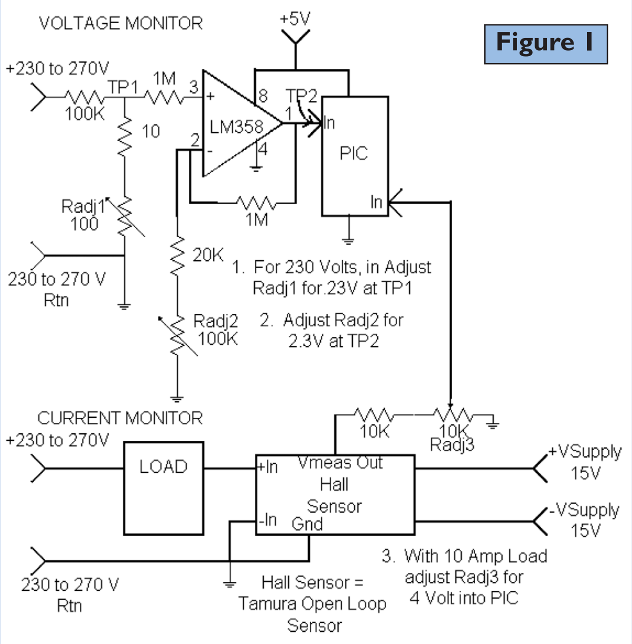
#1 A construction article and schematic can be found in the *ARRL Handbook 2005* starting on page 17.37. It uses the LM723 regulator you wanted. The supply shown is adjustable from 20 to 30 volts, so it satisfies your 24 volt requirement, but is only 10 amps. To modify it for 20 amps, you can halve the resistance of R7 and double its wattage, and double the pass transistor string Q1-Q5 and R2-R5.

Figure 3 (on the next page) shows the addition to the pass transistor string, and the new value for R7.

If you go to 30 amps, you will need to replace the bridge rectifier with one rated over 30 amps, double the value of C1, change R7 to .022

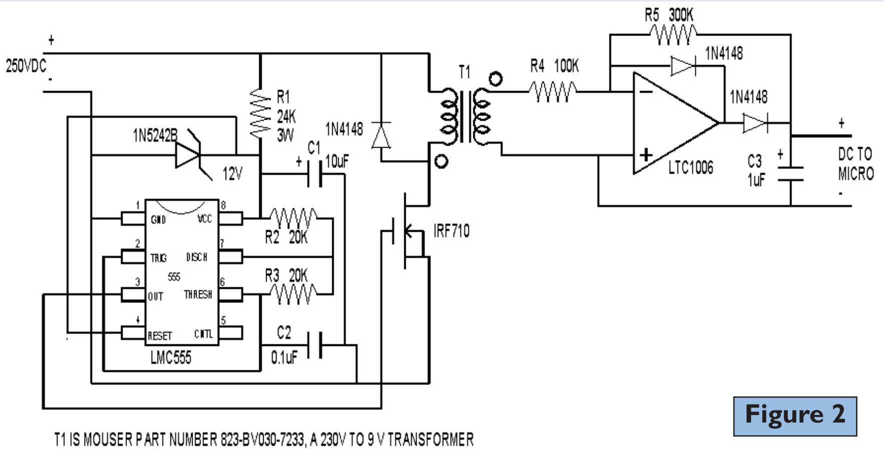
[#7065 - July 2006]

I quite enjoyed reading Russell Kincaid's response on a simple current sensing circuit. Another, tougher problem is on measuring the 250V plus DC voltage. Can Russell or anyone else suggest a circuit that is isolated from the microcontroller?



#1 Monitoring the voltage and current of your 230 to 270 volt array with isolation to the PIC was fairly simple with the voltage monitoring because a simple resistive divider may be used. The main problem is getting a suitable voltage into the PIC so the PIC can give a reliable and stable display. I have found from previous experience the PIC A/D converter works best with voltages between 1 and 4.5 volts. Since the resistive divider furnishes .23 to .27 volts, it is necessary to multiply those voltages by 10 to give 2.3 to 2.7 volts for use by the PIC. A simple eight pin DIP op-amp (LM358) available from Digi-Key was used. Monitoring the current with complete isolation from the PIC requires a Hall sensor device (Tamura Open Loop Current Sensor). That device is also available from Digi-Key (Part # MT7195-ND @ \$14.18). Again, since the sensor output is in volts (i.e., 11 amps = 11 volts out), it is necessary to divide the voltage down to get it within the best range for the PIC. Using the simple resistive divider, 11 volts in gives 4.4 volts out — an ideal range for the PIC. In the programming, you can use some math manipulation to get the correct values displayed for the voltage and current.

Charles Irwin
Hendersonville, NC



T1 IS MOUSER PART NUMBER 823-BV030-7233, A 230V TO 9 V TRANSFORMER

#2 Isolation means a transformer. T1 in this circuit (Figure 2) has a split bobbin so a primary to secondary short is unlikely. The transformer is rated 230 volts primary and 9 volts secondary, 50-60 Hz. The LMC555 is a CMOS type to minimize power. It is an astable running about 250 Hz to provide faster response than if it was running 60 Hz. The secondary voltage is rectified and

filtered for the microprocessor. This has not been built, so some tweaking may be necessary, but the micro can handle the scale factor.

Russell Kincaid
Milford, NH

#3 Avago Technologies (a spin-off of Agilent, which was spun off of HP) makes an "analog" optocoupler, the HCNR200. This is an optocoupler with an LED and two well matched photodiodes, which is designed for isolation of analog signals, primarily for industrial and measurement purposes (scope input isolation). You can find all the necessary information on their website (check for their Optocoupler Designer's Guide and the HCNR200 datasheet). Available at Digi-Key and not expensive.

Walter Heissenberger
Hancock, NH

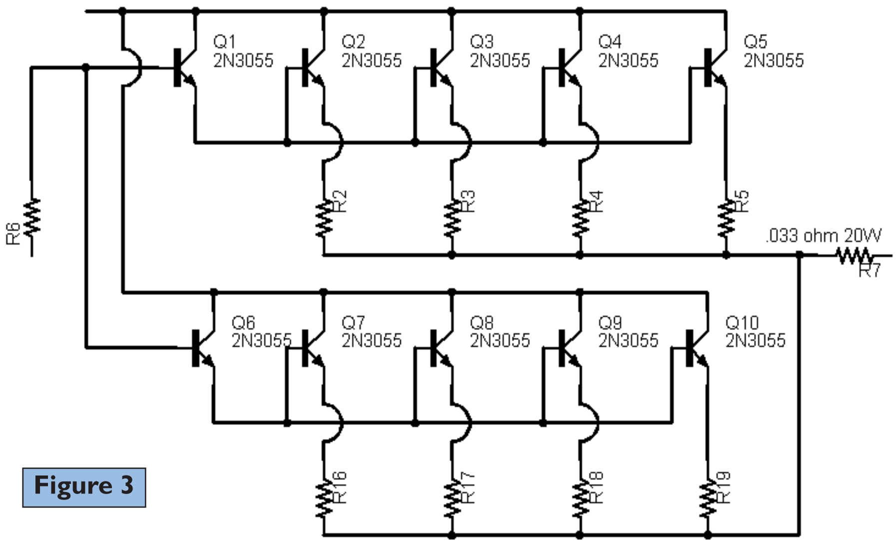


Figure 3

ohms at 40 watts (use two 20 watt resistors), and change Q1 through Q10 to 2N5885 or add a third string of 2N3055s and emitter resistors.

Your transformer will need to be able to deliver at least 1.5 times the current drawn from the supply. To understand that, recognize that when the input voltage (rectified AC) drops below 24 volts, the capacitor provides the difference between the input and the 24 volts delivered to the load. When the input rises above 24 volts, it must re-charge the capacitor ***and*** supply current to the load.

Therefore, the current required from the transformer is greater than the amount the load draws. You also want the AC output from the transformer to be fairly "stiff." If it sags too much under load, regulation will be lost; 5% sag would be acceptable.

Regarding dissipating less heat with MOSFETs vs. bipolar — that is irrelevant in a linear supply. The linear regulates by "throwing away" excess energy as heat. The same amount of excess energy must be dissipated, regardless of the type of transistors used.

Ed Schick
Harrison, NY

#2 The LM723 is so old (nearly 40 years), why do you want to use it? A MOSFET will not save any power unless you use a switching supply. I think you are considering a linear regulator, using the 2N3055 (another ancient device!). This circuit (Figure 4) will supply 30 amps to a 24 volt load. I don't know if you will need an output filter or not, but I think a large value may cause instability. If you are charging a battery, you will not need as large an input filter capacitor. The battery does not care about ripple current. You will have to make L1 using a transformer core, about 500 VA. 800 mH is the minimum. You can make it larger for even better performance.

PARTS LIST

All Mouser parts.

Part	Part number
D1	583-MP3505
R1, R2, R3	284-HS15-0.1
R4	284-HS200-5.0F
R5, R6	71-RH10-0.03
R7	660-MF1/2CC6192F
R8	660-MF1/2CC7151F
Q1, Q2, Q3	511-MJ2955
Q4	511-TIP112
Q5	511-2N3055
C1, C2	75-36DY104F040DC2A
C3	80-C317C472K1R
IC1	511-LM723CN

Russell Kincaid
Milford, NH

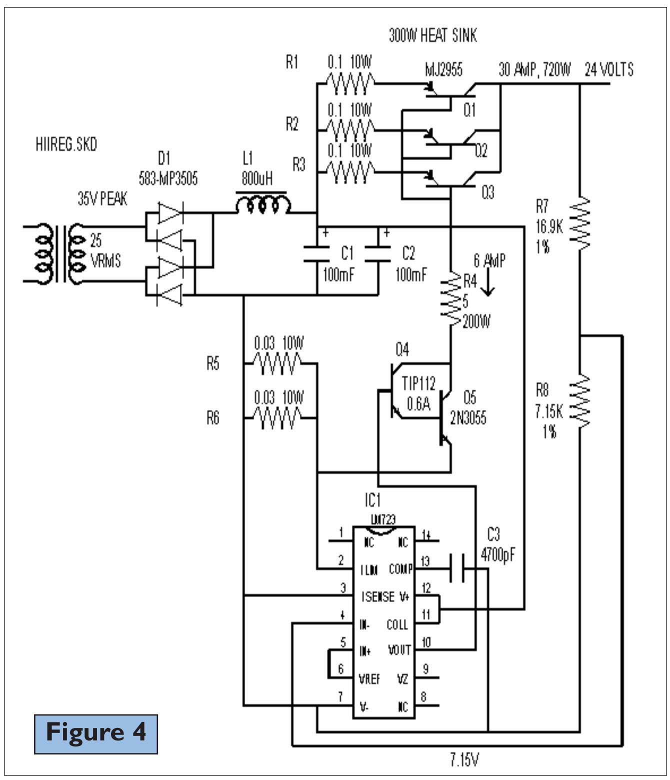


Figure 4

I had requested ideas for an aquarium light timer/controller and had received a couple inventive responses in the last issue. I just wanted to express my appreciation to those who sent in their ideas.

Phil Daniels

■ LOOK FOR ■ SEARCH FOR ■ FIND

Find your favorite advertisers here!

AD INDEX

PowerSupply1 Switching Power Supplies

New to **CircuitSpecialists.com** are these Highly Reliable, Universal AC input/full range single output power supplies. Choose between various 40, 60, 100 & 150 Watt versions. They have the approval of UL and cUL and come 100% full load burn-in tested and are protected with overload/over and voltage/short circuit. Also included is a 2 year warranty.

PowerSupply1	Qty 1	Qty 10	Qty 25	Qty 100	Qty 500	Qty 1000
40W Series Available in 5,12,15,24,48V	$28.99	$26.09ea	$24.53ea	$21.95ea	$15.98ea	$13.79ea
60W Series Available in 5,12,15,24,48V	$32.99	$29.69ea	$27.91ea	$25.95ea	$17.69ea	$15.49ea
100W Series Available in 3.3,5,7.5,12,15,24,48V	$38.50	$34.65ea	$32.57ea	$29.99ea	$21.18ea	$18.49ea
150W Series Available in 5,7.5,9,12,24,28,36V	$48.99	$44.09ea	$39.00ea	$37.50ea	$26.93ea	$23.49ea

Programmable DC Power Supplies

The CSI 3600 Series Programmable DC Power Supplies are equipped with a back-lit LCD display, number keypad and a rotary code switch for ease of use & quick programming. Voltage, Current & Power can all be displayed on the LCD or computer screen (with optional RS-232 interface module). It can be operated at constant current mode, constant voltage mode & constant power mode. It also can be set with maximum limits for current & power output. Ideal instruments for scientific research, educational labs or any application requiring a sophisticated DC-power source.

Model	CSI3644A	CSI3645A	CSI3646A
DC Voltage	0-18V	0-36V	0-72V
DC Current	5A	3A	1.5A
Power (max)	90W	108W	108W

Only $199.00 Each!

www.CircuitSpecialists.com

Programmable DC Electronic Loads

The CSI 3700 series electronic loads are single input programmable DC electronic loads that provide a convenient way to test batteries and DC power supplies. It offers constant current mode, constant resistance mode and constant power mode. The backlight LCD, numerical keypad and rotary knob make it much easier to use. Up to 10 steps of program can be stored.

Model	CSI3710A	CSI3711A
Input Voltage	0-360V DC	0-360V DC
Input Current	0-30A DC	0-30A DC
Input Power	0-150W	0-300W

CSI3710A: $349.00 **CSI3711A: $499.00**

Circuit Specialists Soldering Station w/Ceramic Element & Seperate Solder Stand

$34.95!

- Ceramic heating element for more accurate temp control
- Temp control knob in F(392° to 896°) & C(200° to 489°)
- 3-prong grounded power cord/static safe tip
- Seperate heavy duty iron stand
- Replaceable iron/easy disconnect
- Extra tips etc. shown at web site

Item# **CSI-STATION1A**

Rapid Heat Up!

Also Available w/Digital Display & MicroProcessor Controller

Item# **CSI-STATION2A**

$49.95

Details at Web Site

SMD Hot Tweezer

Adaptor Fits **CSI Stations 1A & 2A and CSI906**

$29.00

Item# **CSITWZ-STATION**

> Soldering Equipment & Supplies > Soldering Stations

In Business

Circuit Specialists Inc.

Since 1971

Heavy Duty Regulated Linear Bench Power Supplies

- Multi-loop high precision voltage regulation
- Automatic voltage & current stabilizing conversion
- Automatic radiant cooling system
- Over-heating protection

CSI5030S: 0-50v/0-30amp **$595.00**

CSI12005S: 0-120v/0-5amp **$595.00**

CSI20002S: 0-200v/0-2amp **$595.00**

Details at Web Site

> Test Equipment > Power Supplies

50,000 Count Advanced DMM w/RS232

Protek

- DC voltage: 1000V
- AC voltage: 750V
- DC accuracy 0.05%
- Low voltage Ohms measurement and conductance
- Auto and software calibration
- Dual back-lit display and K type thermocouple
- Measures duty cycle, pulse width and conductance
- Compare function for go-no go testing
- Peak detection for capturing high speed signals
- Binary bit expression bar graph
- AC and AC + DC true RMS AC voltage and current measurements
- Auto/manual ranging
- Test zener diodes up to 20volts
- Min/Max/Avg relative modes
- Memory store and recall
- Fuse protected current ranges
- Cat II 600V and CE approved

MSRP $199.00

Item# PROTEK608
$129.00

New Lower Price!

Details at Web Site

> Test Equipment > Digital Multimeters

Dual Output DC Bench Power Supplies

High stability digital read-out bench power supplies featuring constant voltage and current outputs. Short-circuit and current limiting protection is provided. SMT PC boards and a built-in cooling fan help ensure reliable performance and long life.

As Low As $93.00!

- Source Effect: $5x10^{-4}$=2mV
- Load Effect: $5x10^{-4}$=2mV
- Ripple Coefficient:: <250uV
- Stepped Current: 30mA +/- 1mA

HOT ITEM!

*All 3 Models have a **1A/5VDC Fixed Output** on the rear panel*

CSI3003X-5: 0-30v/0-3amp 1-4: **$105.95** 5+: **$99.50**

CSI5003X5: 0-50v/0-3amp 1-4: **$114.95** 5+: **$109.00**

CSI3005X5: 0-30v/0-5amp 1-4: **$129.00** 5+: **$121.95**

Details at Web Site > Test Equipment > Power Supplies

Triple Output Bench Power Supplies

with Large LCD Displays

- Output: 0-30VDC x 2 @ 3 or 5 Amps & 1ea. fixed output @ 5VDC@3A
- Source Effect: $5x10^{-4}$=2mV
- Load Effect: $5x10^{-4}$=2mV
- Ripple Coefficient: <250uV
- Stepped Current: 30mA +/- 1mA
- Input Voltage: 110VAC

CSI3003X3: 0-30VDCx2 @3A **$194.50** 5+: **$189.00**

CSI3005XIII: 0-30VDCx2 @5A **$239.00** 5+: **$229.00**

Details at Web Site > Test Equipment > Power Supplies

ESD Safe CPU Controlled SMD Hot Air Rework Station

The heater and air control system are built-in and adjusted by the simple touch of the front keypad for precise settings. Temperature range is from 100°C to 480°C / 212°F to 896°F, and the entire unit will enter a temperature drop state after 15 minutes of non-use for safety and to eliminate excessive wear.

Item# CSI825A++

Sale $149.00!

- CPU Controlled
- Built-in Vacuum System
- Temperature Range:100°C to 480°C / 212°F to 896°F
- 15-Minute Stand-By temperature "sleep" mode
- Power:110/120 VAC, 320 W maximum

Details at Web Site > **Soldering Equipment & Supplies**

Protek 2.9GHz RF Field Strength Analyzer

Fantastic Low Price: $1899.00!

The **3290** is a high quality hand-held RF Field Strength Analyzer with wide band reception ranging from 100kHz to 2900MHz.The 3290 is a compact & lightweight portable analyzer & is a must for RF Technicians. Ideal for testing, installing & maintenance of Mobile Telephone Comm systems, Cellular Phones,Cordless phones, paging systems, cable & Satellite TV as well as antenna installations.May also be used to locate hidden cameras using RF transmissions.

Item# 3290

- WFM/NFM/AM/SSB modulated signals may be measured.
- Signal Levels up to 160Channels can be displayed simulaneously on the LCD
- PLL tuning system for precise frequency measurement and tuning
- Built-in Frequency Counter
- LED Backlight LCD (192x192 dots)
- All fuctions are menu selected.
- RS232C with software for PC & printer interface
- Built-in speaker

(Includes Antenna)

Details at Web Site > **Test Equipment > RF Test Equipment**

Breadboard / Power Supply / MultiFunction DMM Bundle

Provides the user with a quick and efficient system for breadboarding electronic circuits. Comes with three built-in regulated power supplies along with a deluxe, easy-to-use breadboard. Included is a multifunction DMM with 100VDC, 750VAC, frequency, resistance, diode test, audible continuity, transistor check,temperature, and capacitance.

A Super Deal!

Details at Web Site

> **Breadboards & Prototyping Boards**

Powered Breadboard w/out DMM: $69.00

Only $69.99!

Item#: PBB272-DMM: $69.99!

Powered Breadboard w/Multifunction DMM (CSIMS8264)

You Get The DMM for an Extra $.99

Protek Dual Trace 100MHz Oscilloscope

Outstanding performance and durability for an incredibly low price of $519. You will find it at most other locations selling for $975.

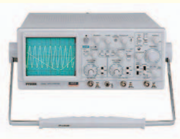
- Four traces may be simultaneaously displayed in ALT-sweep
- Five vertical Modes Chl, Ch2, Dual, Add and Subtract
- Bright 6" CRT with an internal graticule
- 12 kv acceleration voltage
- Sweep speeds to 2nS/Div.

Item# 6510

Limited Offer $519.00!

Details at Web Site > **Test Equipment > Oscilloscopes/Outstanding prices**

Protek 20MHz Dual Trace Oscilloscope



- 20MHz Bandwidth
- Alt-Mag sweep for simultaneous display of main and X5 magnified trace
- 1mV/Div Vertical sensitivity
- Alternate trigger for a stable display of unrelated signals
- Multi-level trigger
- X5 Sweep Magnification

Details at Web Site

$279.00!

> **Test Equipment > Oscilloscopes/Outstanding prices**

Protek 2MHz Sweep Function Generator

- 0.02Hz-2MHz(7 Ranges)
- Sine,Triangle,Square,Pulse,Ramp, Slewed Sine Waveform
- Sync. Out (TTL Square Waveform)
- Accuracy: ±5% of Full Scale to 200KHz, ±8% of Full Scale from 200KHz-2MHz
- Sweep Function
- VCG Input

Item# PROTEK 9205

Special $159.00! Any Quantity

Details at Web Site

> **Test Equipment > Function Generators**

Digital Storage Oscilloscope Module

PC based Digital Storage Oscilloscope, 200MHz 5GS/s equiv. sampling USB interface

Convert any PC with USB interface to a high performance Digital Storage Oscilloscope. This is a sophisticated PC basedscope adaptor providing performance compatible to mid/high level stand alone products costing much more! Comes with two probes.

Details & Software Download at Web Site

Price Breakthrough!

> **Test Equipment > Oscilloscopes/Outstanding Prices**

Item# 200DSO Only$749.00

SONY Super HAD CCD Color Weatherproof IR Cameras

480 TV Lines Resolution

- Day & Night Auto Switch
- Signal System: NTSC
- Image Sensor: 1/3" **SONY Super HAD CCD**
- Horizontal Resolution: 480TV lines
- Min. Illumination: 0Lux

Item# VC-827D: $132.00

SONY Super HAD CCD™ equipped camera's feature dramatically improved light sensitivity

Intelligent DMM w/ RS-232

- 3999 Counts and 38 Segment Bar Graph
- Dual Display (digits & bar graph)
- Capacitance Function, Transistor & Diode Test
- Frequency Range & Temperature
- RS232C Standard Interface
- Data Hold

Item# CSI345

SALE $29.95

Details at Web Site

> Test Equipment > Digital Multimeters/World Beater Prices

SONY Super HAD CCD B/W Weatherproof IR Camera

- Day & Night Auto Switch
- Signal System: EIA
- Image Sensor: 1/3" **SONY Super HAD CCD**
- Horizontal Resolution: 420TV lines
- Min. Illumination: 0Lux

Item# VC-317D: $59.50

Details at Web Site

SONY Super HAD CCD Color Camera

- Weather Proof
- Signal System: NTSC
- Image Sensor: 1/4" SONY Super HAD CCD
- Horizontal Resolution: 420TV lines
- Min. Illumination: 1Lux/F1.2

Item# VC-805: $53.95

> **Miniature Cameras(Board,Bullet,Mini's, B/W, Color)**

SONY Super HAD CCD Color Weatherproof IR Camera

- Day & Night Auto Switch
- Signal System: NTSC
- Image Sensor: 1/4" **SONY Super HAD CCD**
- Horizontal Resolution: 420TV lines
- Min. Illumination: 0Lux

Item# VC-819D: $62.50

Visit our website for a complete listing of our offers. We have over 8,000 electronic items on line @ **www.CircuitSpecialists.com**. PC based data acquisition, industrial computers, loads of test equipment, optics, I.C's, transistors, diodes, resistors, potentiometers, motion control products, capacitors,miniature observation cameras, panel meters, chemicals for electronics, do it yourself printed circuit supplies for PCB fabrication, educational D.I.Y. kits, cooling fans, heat shrink, cable ties & other wire handleing items, hand tools for electronics, breadboards, trainers, programmers & much much more! ***Some Deals you won't believe!***

Circuit Specialists, Inc. 220 S. Country Club Dr., Mesa, AZ 85210
800-528-1417 / 480-464-2485 / FAX: 480-464-5824